A Simple Approach to Synthetic Time Traces Generated from Basic Parameters

av

HUI JIN

MASTEROPPGAVE

for graden

Master i Anvendt matematikk og mekanikk

(Master of Science)

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

June 2013

Faculty of Mathematics and Natural Sciences
University of Oslo

Contents

Chapter 1

Introduction to the Thesis

1.1 Motivation of the Thesis

High-fidelity multiphase flow models of the well hydraulics and flow lines in production and drilling are used to simulate and test control systems. Examples of such application are multiphase flow in oil production managed pressure drilling, under-balanced drilling and automated well control in drilling, where automatic control systems are used to stabilize pressure or flow in various ways.

A discrepancy of existing high-fidelity models, is that they do not reproduce qualitatively realistic colored noise as that caused by various flow phenomena, such as waves, hydrodynamic slugging and other unmodelled stochastic flow dynamics. This means the control systems cannot be tested properly for robustness with respect to realistic colored noise by simulations, which is a fundamental limitation in the development and testing of control systems for these applications.

The possibility in render stochastic colored noise with realistic properties to add as noise to existing hydraulic flow models, e.g. based on a measured time-series, will fundamentally increase the value of simulations with respect to assessing robustness of control systems by simulations. This will serve as a valuable tool in the design of control systems for flow applications.

1.2 Specification of the Thesis

The objective of this thesis will be focused on generating synthetic time traces from the liquid fraction measurements in two phase high pressure pipe flow. The measurement data which serves as a foundation for this thesis work is provided by Associate Professor George William Johnson at

University of Oslo. The main objective of this work is to develop an algorithm which could be conveniently and easily used to generate synthetic time traces of stratified wavy flows. The main focus of the thesis is then to develop a robust algorithm with only simple programming techniques along with as few input variables as possible, to generate the synthetic time traces which approximately matches the ones from the real experiments. Therefore, complicated techniques and advanced toolboxes are mostly avoided in the work. After the codes are developed, comparisons between synthetic time traces with the measurements will be made using power spectral density, statistical properties and other methods. Afterwards, the parameters in the codes will be modified in order to achieve the best approximation. The synthetic time traces should be useful to provide realistic input of approximations to simulations of the control systems described in 1.1

1.3 Programming Software

MATLAB[1] is used throughout the thesis. As described above, only normal programming techniques are used and no advanced toolboxes are called for. Therefore, it is very efficient to get the desired figures for any given data for the experiment measurements in the thesis work.

1.4 Organization of the Thesis

This thesis is divided into four chapters with 4 appendices. The first chapter is the introduction and the goal of the thesis work. Chapter 2 is the brief introduction to the theoretical foundation of the thesis, which includes the idea of fractals, the fundamental knowledge in mathematical statistics and digital signal processing. The approach is given in Chapter 3 with results and comparisons between the synthetic time traces and real experiment measurements. The final conclusion is made in Chapter 4. In addition, there are several comprehensive appendices which include the figures, tables and program codes.

[1] version R2009a

1.5 Notations

The Theoretical Symbols

A	Mean Value
σ	Standard Deviation
σ^2	Variance
A_p	The Average of Polynomial Function

The Program Code and Experiment Related Notations

PN	Polynomial Degree
HEIGHT	Wave Height
TROUGH	Trough Height
w	Number of Generations
z	Number of Synthetic Single Waves
Max	Maximum Wave Height
Min	Minimum Wave Height
H_ω	Characteristic Wave Height
MP	The Power at Major Frequency
Freq	Major Frequency
NBox	Equals z, the Number of Synthetic Single Waves
k,cc	Column Number
dn, cn	File Names for Error Estimates' Table
l	Mean Value of the Experimental Measurement Data
m	Standard Deviation of the Experimental Measurement Data

Other Terms of Interests

Resolution	Number of Points for Resulting Waves. Matching the Resolution is critical.
ΔT	The Sampling Frequency of the Experimental Measurement Data at 82Hz.
Err	The Error Function for Computing the Error Estimates.
$R(N_r)$	The Result after the time traces were sampled.
$S(N_r\vert \sim)$	The Sampling Function (Taking points of z+1 intervals of points)
N_i	The Number of Total Points after Generation of Wave Trains

Chapter 2

The Theoretical Foundations

The main idea of the thesis is just to apply the simplest approach available to obtain desirable results. The theoretical foundation of the thesis lies in the idea of fractals, elementary definitions in mathematical statistics and basic digital signal processing. The following sections will offer detailed information regarding these used in the thesis work and code development.

2.1 The Idea of Fractals

What are the fractals? Before we start with the very neat definition given by Mandelbrot, let's first understand the characteristics of the geometry of fractals from an informal point of view. Fractal geometry is the study of the form and structure of rough and irregular phenomena. Fractals are **sets** defined by the three related principles of *self-similarity*, *scale invariance*,and *power law relations*. When these principles converge, fractal patterns form. [BL p.2]

2.1.1 Fractal Geometry, a Natural Phenomenon

Fractal geometry is nearly everywhere in the world around us from a branch of tree twig to the snowflakes. The simplest illustration of fractal geometry is the figure below. It certainly shows the complexity of the natural phenomenon and also contains some typical characteristics if we examine the figure closely.

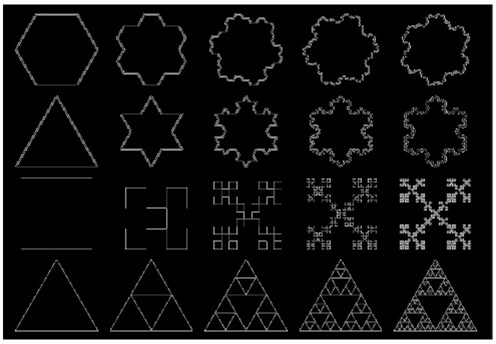


Figure 2.1: *A Figure from Mathworld*

2.1.2 Characteristics of Fractals

Sets

Fractals are defined as sets. Any kind of data set can be considered as fractals. Thus, points, lines, surfaces, multidimensional data and time series are in fact fractals. Fractal geometry is very common in our world. In the thesis, the measurement data is actually a set of time series.

Self-Similarity

An object is self-similar when it is composed of smaller copies of itself, and each of those smaller copies is in turn made up of yet smaller copies of the whole, and so on.The word similar also carries the geometrical meaning: objects that have the same form but may be different in size. The result is an object composed of a single pattern that repeats itself many times at many different sizes. Moreover, as their shrinks, the copies of the pattern multiply, so the smaller the size the greater the frequency of the copies. Conceptually, the process is iterative: At each scale, the construction process repeats itself.

Scale Invariance

Self-similarity entails scale invariance, so scale invariance is also diagnostic of fractals. A thing is scale invariant when it has the same characteristics at every scale of observation. As a result, if you zoom in on a fractal object, observing it at ever-increasing magnification, it still looks the same. The relationship to self-similarity is direct and inevitable. Because a self-similar object is composed of copies of copies of itself at every scale, it looks the same at every scale of observation.

Again, as with self-similarity, only perfect mathematical objects look exactly the same at every scale of observation. Real objects usually exhibit scale invariance statistically within the finite size limits. For example, many geological and geographical phenomena exhibit fractal structures, but they cannot be scale invariant at sizes greater than the Earth or smaller then the molecules composing the rocks or soils.

Power Law Relations

Self-similarity implies a type of relationship called a "power law". For a set to achieve the complexity and irregularity of a fractal, the number of self-similar pieces must be related to their size by power law. The connection between power laws and fractals is deep and intimate. Power law distributions are scale invariant because the shape of the function is the same at every magnitude. Power law distributions are the only scale invariant distributions.

2.1.3 The Formal Definition

Mandelbrot offers very precise definition of what a fractal is in his renowned book.

> *A fractal is by definition a set for which the Hausdorff-Besicovitch dimension strictly exceeds the topological dimension (page 15)*

This definition can assist us when deciding if a complex pattern is indeed belonging to the geometry of fractals even if the pattern fulfils the three main characteristics.

2.2 Elementary Mathematical Statistics

The following statistical terms are used in the work: mean value, standard deviation and variance. Although as an advanced program software, MATLAB provides all the necessary ready-made functions, it is still helpful to review the definitions of the terms and their intuitive meanings.

2.2.1 Mean Value

The mean value, or often called arithmetic mean is the central tendency of a collection of numbers taken as the sum of the numbers divided by the size of the collection. While the arithmetic mean is often used to report central tendencies, it is not a robust statistic, meaning that it is greatly influenced by outliers. Notably, for skewed distributions, the arithmetic mean may not accord with one's notion of "middle", and robust statistics such as the median may be a better description of central tendency.

Definition 2.1 *Suppose we have sample space* $\{a_1, ..., a_n\}$ *then the mean value is defined as* $A = \frac{1}{n}\sum_{n}^{i=1} a_i$

If the list is a statistical population, then the mean of that population is called a population mean. If the list is a statistical sample, we call the resulting statistic a sample mean.

2.2.2 Expected Values

The concept of the expected values is rather similar to the notion of weighted average. The possible values of the random variable are weighted by their probabilities. It is often helpful to think of the expected value as the center of mass of the frequency function.

Definition 2.2 *If* X *is a discrete random variable with frequency function* $p(x)$, *the expected value of* X, *denoted by* $E(X)$ *is* $E(X) = \sum_i x_i p(x_i)$

The expected value is not used in the thesis work; instead it serves as a building block to define the next two statistical terms.

2.2.3 Variance and Standard Deviation

The expected value can be view as an indication of the central value of the density or frequency function. It is sometimes regarded as a location parameter. In addition, the median of a distribution is also a location parameter and is not necessarily equal the mean. Therefore, another parameter is needed, the standard deviation, which shows how dispersed the probability distribution is about its center, of how spread out on the average are the values of the random variables about its expectations. Before doing that, we need to define the **variance**.

Definition 2.3 *If* X *is a random variable with expected value of* $E(X)$, *the variance of* X *is* $Var(X) = E[X - E(X)]^2$ *provided that the expectation exists. The standard deviation of* X *is the square root of the variance.*

If X is a discrete random variable with frequency function $p(x)$ and expected value $\mu = E(X)$, then $Var(X) = \sum_i (x_i - \mu)^2 p(x_i)$ whereas if X is a continuous random variable with density function $f(x)$ and $E(X) = \mu$, then $Var(X) = \int_{\infty}^{-\infty} (x - \mu)^2 f(x) dx$

The variance is often denoted by σ^2 and the standard deviation by σ. The variance of a random variable changes in a simple way under linear transformations.

2.3 Digital Signal Processing

In this work, the idea of sampling, fast Fourier transform and power spectrum are needed.

2.3.1 Sampling

Sampling, simply put, is to take points at constant time intervals. In the experiment data collection process, the sensor projects signals downward to the interface where both liquids interact at a frequency at 82Hz, that is, the signal collection is done 82 times in one second.

In addition, when the points of interest exceed what we need, we also need to take sampling process to reduce the points in order to match the ones we want by the same process stated above by using MATLAB. This process,in this work, is called resolution adjustment.

2.3.2 Fast Fourier Transform & Power Spectrum

Fourier Transform is used to analyse the time traces by displaying the power along the frequency axis. In MATLAB, ready-made Fast Fourier Transform is used in order to efficient find the power along the frequency axis of any given experimental measurement data.

When this process is done, we can continue to compute the power distribution given by the formula $p = |(fft(x)/(N-2))|$; where $fft(x)$ is the Fast Fourier Transform function, N is the unit step. Afterwards, we can plot the power distribution along the frequency axis to get the values on which the power of the signals mainly lies, that is, the power spectrum. Power spectrum tells us the strength of the signals at various frequencies. The logarithmic frequency axis method shows the major power on the corresponding major frequencies.

Chapter 3

The Approach,Program Code Development and Implementation

3.1 The Approach

The approach is based upon the idea provided by Prof. Johnson. This author does not develop a new approach; rather, a modified method following the track of the original approach.

3.1.1 The Essentials by Prof. Johnson

This thesis work is based upon the program codes and method supplied by Prof. Johnson, by which this author tries to build up a more "coherent" working code to obtain the desired figures automatically. Thus, the focus of this thesis mainly lies in the comparison with the experiment measurements data, that is the evaluation of the method and its validity.

The Initial Codes, Parameters and the Method

Prof. Johnson supplied two working codes which can generate a single synthetic time trace and a train of synthetic time traces. The two original codes are listed in the appendix. The codes involve altering the following parameters: *PN (Polynomial Degree),HEIGHT (Wave Height), TROUGH (Trough Height),w (Number of Generations)*z (Number of Single Synthetic Waves).Since the initial codes do not take account of experiment measurement data, the main focus of the original codes is to find the relations between *PN (Polynomial Degree) and w (Number of Generations).*

The method, briefly described, is to first create a single synthetic wave. Then single synthetic wave is pieced together by certain numbers to be formed into a synthetic wave trains which will be modified to match the experiment measurement data.

The Single Synthetic Wave

The core of the method lies in the code FL2.m. The code can generates the single time traces with varying degrees of PN (Polynomial Degree) and w (Number of Generations). The first step is to find the relations between them by changing the values of the two parameters systematically. e.g. taking PN from 1 to 7 and w from 1 to 12 as illustrated in the following example.

1. ***The "prototype" Time Trace with 1 Generation***

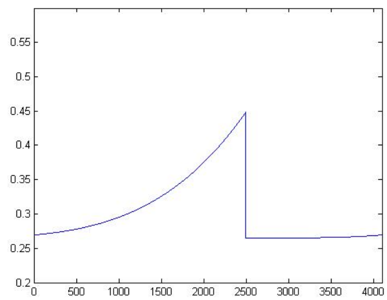


Figure 3.1: *The Single Time Trace with Polynomial Degree of 4 and Number of Generation of 1*

2. ***The Single Time Trace with 3 Generations***

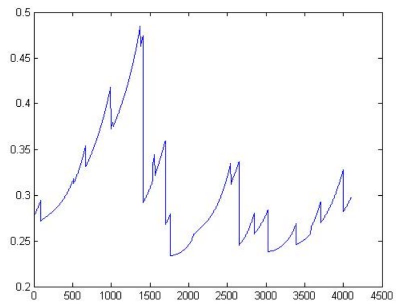


Figure 3.2: *The Single Time Trace with Polynomial Degree of 4 and Number of Generation of 3*

3. ***The Single Time Trace with 6 Generations***

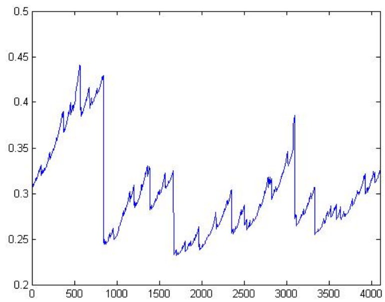


Figure 3.3: *The Single Time Trace with Polynomial Degree of 4 and Number of Generation of 6*

4. ***The Single Time Trace with 12 Generations***

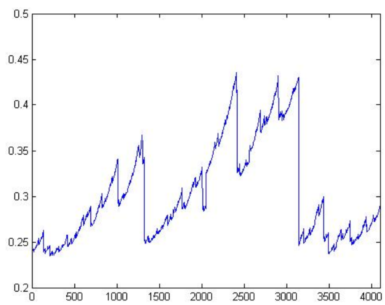


Figure 3.4: *The Single Time Trace with Polynomial Degree of 4 and Number of Generation of 12*

As we can see it clearly that the number of the generations will greatly affect the single time trace. Furthermore, the degree of polynomials also visibly affect the traces as well as they are shown below.

1. ***The Single Time Trace with Degree of Polynomial 1***

2. ***The Single Time Trace with Degree of Polynomial 4***

3. ***The Single Time Trace with Degree of Polynomial 8***

We can see that the larger the degree of polynomial become, the single time trace tends to be flattened out. Therefore, it is critical to choose the correct degree of polynomials and number of generations. In this thesis work, the polynomial degree is set from 1 to 10 while number of generations is from 1 to 12, which provides enough approximation of results we desire.

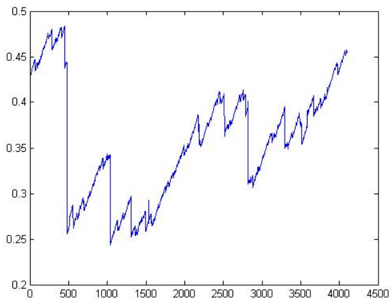


Figure 3.5: *The Single Time Trace with Polynomial Degree of 1 and Number of Generations of 9*

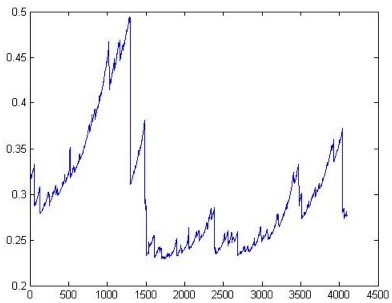


Figure 3.6: *The Single Time Trace with Polynomial Degree of 4 and Number of Generations of 9*

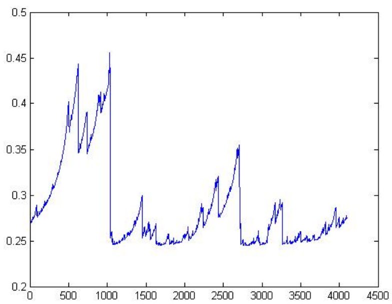


Figure 3.7: *The Single Time Trace with Polynomial Degree of 8 and Number of Generations of 9*

The Synthetic Wave Trains

After a single synthetic wave is determined with the parameters, it is then necessary to calculate the number of single synthetic waves in order to construct a synthetic wave train. The key lies in the power spectrum of the experiment measurement data. The assumption is if the synthetic wave train resembles very closely to the experimental measurement data, thus, its power spectrum should be quite similar as well. In another word, by analysing the major frequency of the experiment data, we can find out the exact numbers of single synthetic wave needed for the train.

After a fast Fourier transformation has been done to the experimental measurement, the major frequency at which the biggest value of the power locates can be found. Since the frequency of the experiment measurement data is taken at 82Hz. Therefore, the multiplication of the major frequency value with the measurement frequency (82Hz) will yield the number of the single synthetic waves which would be formed into the wave train we desire. However, because of the multiplication, the resolution of the newly obtained wave train will actually have more points than the experimental ones. Thus, it is necessary to take the points out of every certain intervals as a sampling process so that we can obtain the wave train which has the same resolution to the experimental one.

The Equations

There are some essential equations which are incorporated into the program codes. The equations are given in their original forms. While being incorporated into codes, modifications are sometimes made.

1. Polynomial Function
   $f(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$

2. The Average of a Polynomial
   $A_p = \frac{1}{\Delta x}\int x^n dx$

3. Wave Height
   $H_\omega = 2\sqrt{2}\sigma$

4. Trough Height
   $h_e = \frac{1}{\Delta X}\int(H_w X^n + h_t)dx \Longrightarrow h_t = h_e - \frac{H_\omega}{(n+1)}$

Some detailed explanations about the equations: σ stands for the standard deviation from the experiment, h_e is the mean liquid height from the experiment, H_ω is the wave height, H_t is the trough height. A detailed description is listed in the notations.

3.2 The Analysis of Experiment Measurements

3.2.1 The Experimental Measurement

The Experiment is a Real Process

The measurement data is collected via real life experiment in the laboratory. The two-phase gas liquid pipe flow is often being considered with a specific focus on large breaking waves. The whole data is not artificially made; instead, the real life experiment data can actively serve the purpose the work that the approach is applicable in most real cases.

Some Properties of the Experiment

The experiment is taken at a sampling frequency of 82Hz in gamma densitometers. The two-phase pipe flow is consisted of heavy oil and water, with both moving at very fast speed. The pipe line is 20 meters long, the diameter of the pipe is 1 meter and the inclination is zero. The point of the interest is in the boundary surfaces where the heavy oil and water interact, in which the wavy flow with large breaking waves form.

3.2.2 The Parameters Collected from the Experiment

The focus of the work is the comparison between the experimental measurement and the synthetic wave train. Therefore, it is necessary to collect some critical values from the experiments to be later used as input parameters and run the code with the right choices of the parameters. All the critical values can be obtained in MATLAB easily by using the built in function and the equations aforementioned. The following is an example of an analysis of the experiment data which contains six columns of measurements for each run of the experiment. A brief notation is also included.

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3106	0.0647	0.1830	0.6576	0.1719	0.0054	0.5122	42.0000
0.2802	0.0701	0.1983	0.6329	0.1372	0.0082	0.5122	42.0000
0.2971	0.0698	0.1974	0.6030	0.1239	0.0081	0.5122	42.0000
0.3105	0.0703	0.1988	0.6763	0.1553	0.0083	0.5122	42.0000
0.3012	0.0685	0.1937	0.6842	0.1657	0.0053	0.5122	42.0000
0.3038	0.0752	0.2127	0.6734	0.1858	0.0100	0.5122	42.0000

Figure 3.8: ***A Sample Table Obtained after Simple Calculations are Done***

NOTATIONS

Mean	Mean Value
STD	Standard Deviation
H_omega	Wave Height
Max	Maximum Wave Height
Min	Minimum Wave Height
MP	Maximum Power at the major frequency.
Freq	The Major Frequency
Nbox	The Number of Single Synthetic Time Traces

3.3 The Description of the Process

Based on the initial codes, a more "coherent" code was made in order to create figures with the best match. This section contains two parts. The first part is to introduce the input parameters essential to the code and the second part is the detailed explanation of the algorithm along with some central formulas.

3.3.1 The Input Parameters

There are two types of the input parameters. One is the essential parameters which serves as a core in obtaining the results. The other type is the utility parameters which offers easier views when comparing the results and creating labels etc.

The Essential Parameters

These are the critical parameters which lays the foundation for the codes to generate a batch of figures. They will be written in the same way in the code.

x is the wave height (H_ω from the experiment). z is the number of boxes (Each box contains one synthetic single wave).t is the polynomial degree while w is the number of generation. In this work, x,z is calculated by using the experimental measurement data as fixed values. t and w are varying values; and with each pair the code generates a figure and corresponding error table. t is set from 1 to 7 while w 1 to 12 in the work.

The Utility Parameters

These parameters serve as support roles in comparing and labelling the figures and tables. k and cc are the column number of the experiment; dn is the measurement data names, l,m stand for the mean and standard deviation from the experimental measurement data.

3.3.2 The Algorithm

The following section will in detailed describe the process of the code running and some formulas will be included to offer an easier view to the approach.

1. ***STEP ONE: Analyse the Experiment Measurement***
   Obtain the necessary input parameters, e.g. H_omega, Mean Value, Standard Deviation, Number of Single Synthetic Time Traces via the multiplication of power spectra and sampling frequency. It is very straightforward to obtain mean value and standard deviation from MATLAB. The formula for obtaining z (The Number of Single Synthetic Time Trace) is the following:
   Let ΔT be the sampling frequency with a value of 82. The major frequency $Freq$ can be obtained from the function. Thus we can get $z = Freq * (\Delta T)$

2. ***STEP TWO: Create A Relevant Single Time Trace***
   The initial code,in short, uses the "polynomial function" $f_0 = F(a^n)$ with the right number of generations w, where w represents the "holder" of how many such function like f_0 should be combined to generate one single synthetic wave. In addition, the wave height H_ω obtained from the experimental measurement data, will decide the right characteristic group of figures which behave like "boundary conditions".

3. ***STEP THREE: Generate A Whole Wave Train***
   This step involves generating a synthetic wave train. After the single synthetic wave is created, the approach is to piece together all the single waves to create a synthetic wave train, $F_T : F_T = f_0+f_1+\cdots+f_n$ where f_n is the number of z, the number of single wave needed for such a wave train.

4. ***STEP FOUR: Sampling and Error Estimates***
   Each of the synthetic wave like f_0 will contain as many points as in the experimental measurement data. Therefore, the overall points of F_T will greatly exceed the resolution from the experimental measurement. Sampling is then essential to scale down the time traces so that further comparison can be made. The sampling result can be expressed as $R(N_r) = S(N_r|1 : (z + 1) : N_i)$ where N_r is the points which match the experimental measurement and N_i is the total points from the

synthetic wave train before sampling.In short, it shows the code takes points at certain intervals $(z + 1)$ to get the resolutions match each other.

The last step for the code to do is to create a table and a group of corresponding figures. The error estimates of mean values and standard deviation between the experimental measurements and synthetic wave trains will be produced. Each pair of such an estimate will also associate a figure.Error Estimate Formula $Err = (E_0 - E_1)/100 * E_1$

5. ***STEP Five: Read the Table and Get the Figure***
   After the code is completely run, a table with error estimates and corresponding figures will also be created. The final stage is the approximation process. The assumption is that if the synthetic time wave trains match the experimental measurements. These should also produce nearly the same values in mean values and standard deviation. Thus, the pair with the least error estimates is the one we are interested in. But checking up the polynomial degree and number of generation, we can also find the corresponding figure associated with the pair in a group of figures generated with varying polynomial degrees, number of generations.

For more details, please refer to the complete MATLAB code in the appendix.

3.4 An Illustrating Example

The following is an example for generating such a synthetic wave train. An experimental measurement data will be used[1].

3.4.1 Parameters and the Table

The first thing to do is to insert values into the parameters for the main function code. [ox(x,k,z,cc,dn,cn,l,m) $\longrightarrow$ x=H_omega, k,cc=COLUMN, z=Nbox, dn=Set No. cn=Column No, l=mean value, m=standard deviation]. The values of parameters are all fixed from the experiments except the degree polynomial degrees and number of generations. Since the code is designed to be run from polynomial degree 1 to 7 and number of generations

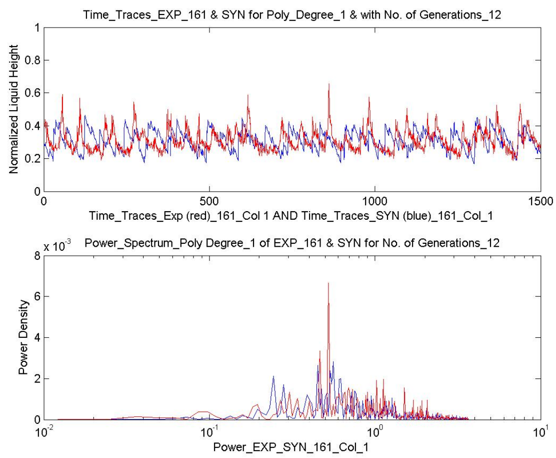
[1] EXOct22_130914_rn_161_w_0.462_o_0.000_g_3.393_bi_0.10_RW_

from 1 to 12. Then it will generate a table with 84 pairs error estimates along with 84 figures which show the comparison between the experimental measurement and synthetic wave trains, and the power spectrum comparison.

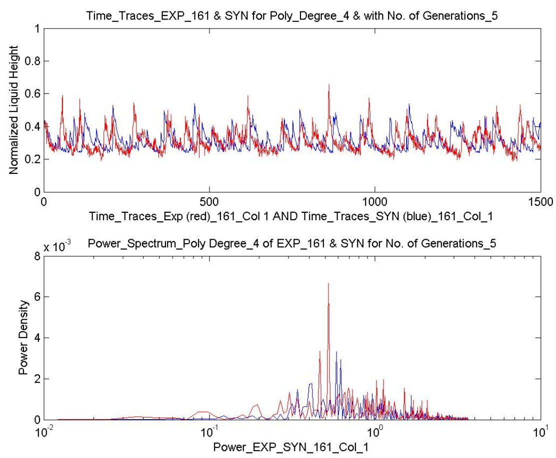
3.4.2 The Corresponding Figures

The code generates figures. Because of limited space, only some will be shown randomly. These will be shown to represent the wide variations of the plots in the figures.

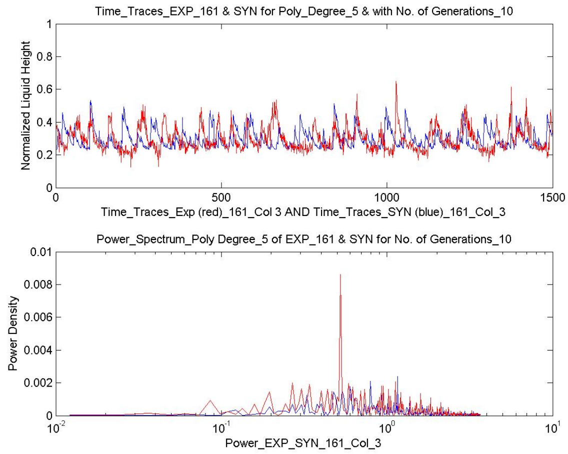
1. *Randomly Chosen Figure No.1*



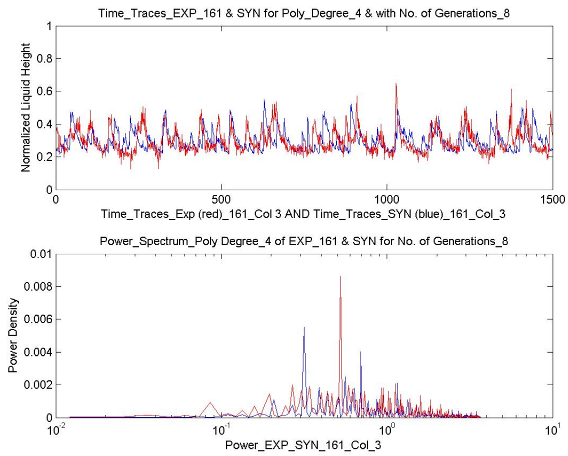
2. *Randomly Chosen Figure No.2*



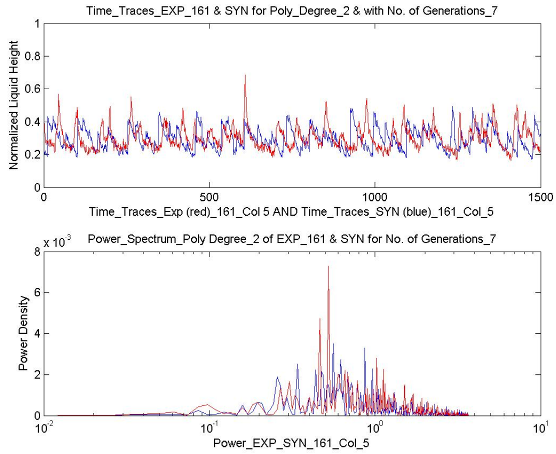
3. *Randomly Chosen Figure No.3*



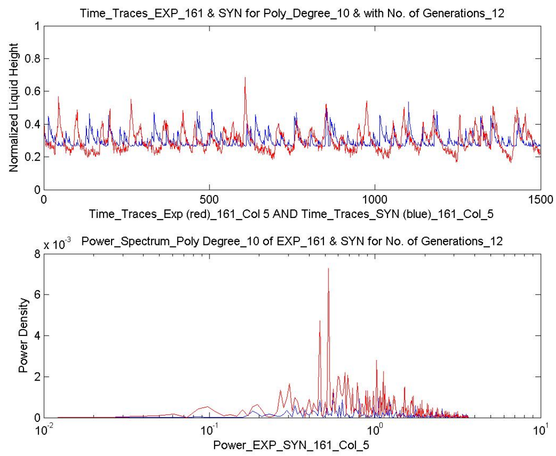
4. *Randomly Chosen Figure No.4*



5. *Randomly Chosen Figure No.5*



6. *Randomly Chosen Figure No.6*



3.4.3 The Figures of Good Approximations

Our assumption for the thesis work is that if we can create the synthetic wave trains which have the same shape as the experimental measurement data. Then the error estimates should be as small as possible since they would share the most of the statistical properties as well. Therefore, by reading the error estimate pairs in the table, we can search for pair with the least error then find the corresponding figures.

COMMENTS: Compared to most of the rest figures, these figures match the experiment measurements better. However, if close reading from the table, one can find out that there are several sets of numbers which are close to each other in values. And that will result in creating several similar figures. Therefore, in this work, the term arbitrarily' is emphasized. If more accurate figures are needed, more advanced techniques are necessary.

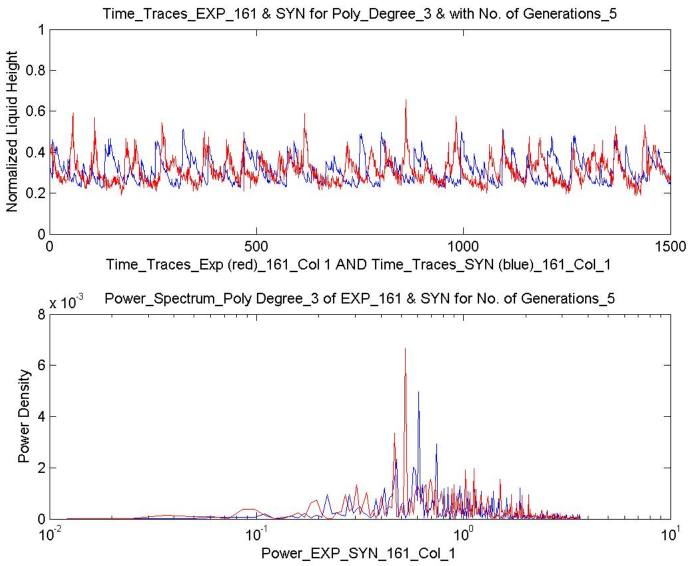


Figure 3.9: ***The Figure For Column One***

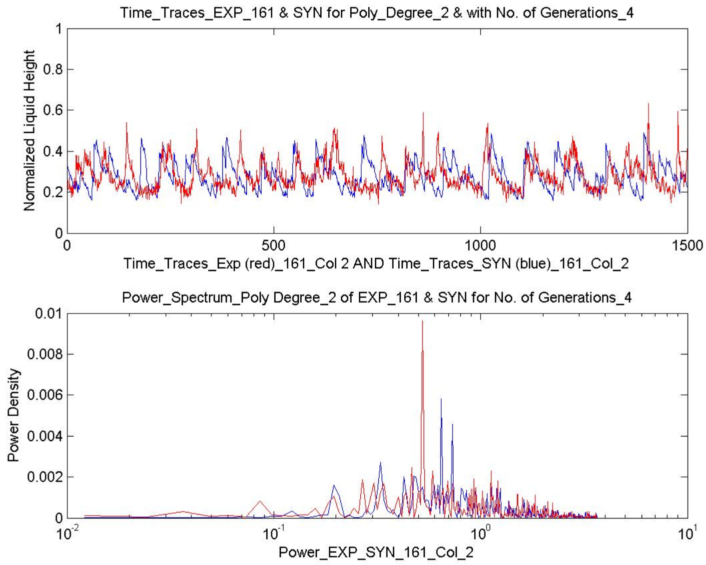


Figure 3.10: ***The Figure For Column Two***

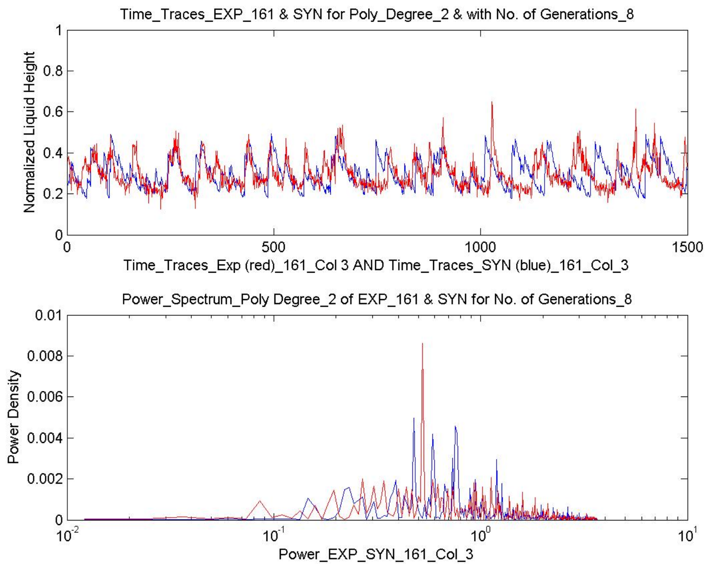


Figure 3.11: ***The Figure For Column Three***

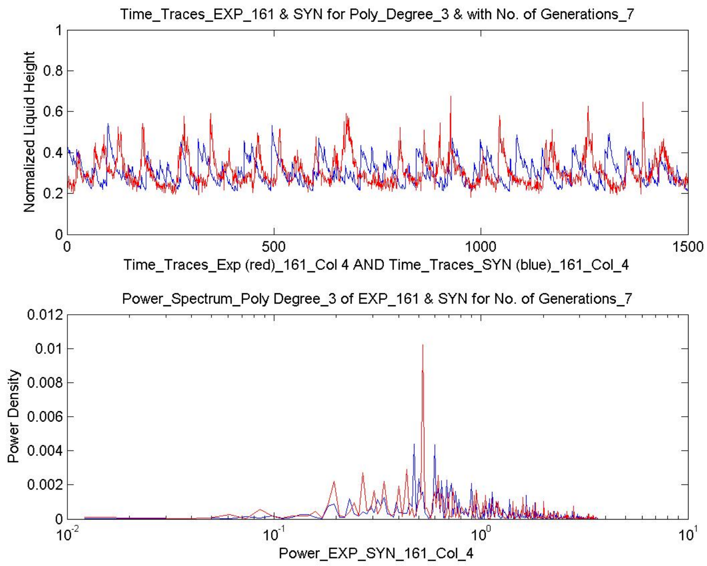


Figure 3.12: ***The Figure For Column Four***

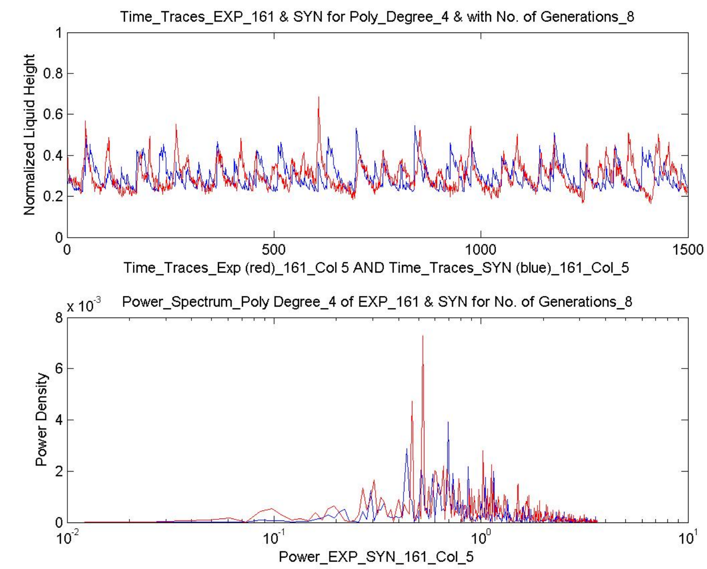


Figure 3.13: ***The Figure For Column Five***

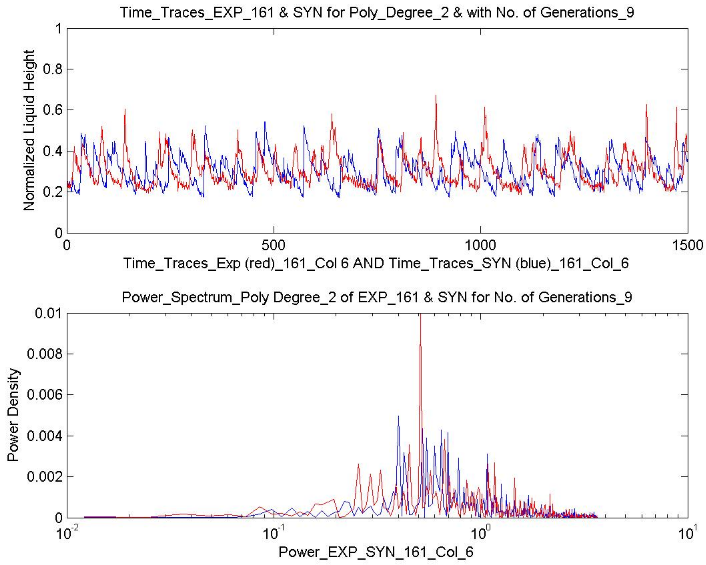


Figure 3.14: ***The Figure For Column Six***

Chapter 4

Summary and Conclusion

4.1 The Summary of the Thesis work

The goal for the thesis is to develop an algorithm with only the simple functions and techniques to create a realistic synthetic time traces very fast by easily obtainable parameters from the experimental measurement data. MATLAB is used for the work.

The work starts with the assumption that if the synthetic wave trains match the experimental measurements. Then they must share the similar statistical properties. The initial stage is to analyse the experiment data, which will yield fixed values of mean values, standard deviation. Then by using the essential equations, wave height, major frequency, etc. are also available.

The next step is by inserting the parameters into the code then run the code repeatedly for each value of polynomial degree and number of generation as there are correlating relations between the two unfixed variables. Thus, the approach covers all the available outcomes. As it is shown in the section of the illustrating example, the varying number for polynomial degree is 1 to 7 and 1 to 12 to number of generations. And all the combination of the two variables were considered and complete table and figures were made. Therefore, the approach systematically evaluated all the available parameters for given choice of data.

The last step is to read through the table of error estimates to find the pair which gives the least error. Then it is possible to easily locate the right figures. Since some pairs share almost the same least errors, it should be noted that we can "arbitrarily" choose the figures based upon the reading.

4.2 The Evaluation and the Conclusion

The method is very efficient in modelling the real physical waves because only simple functions and techniques are used throughout the program. Thus, the computing speed is very fast. In addition, the method covers all the varying parameters and provide quite resemblances between the measurement data and synthetic wave trains. As shown in the complete figures in the appendix for the best matches, the method does provide reliable approximations between relevant figures. Therefore, it serves well for the aim of the work in the introduction section.

The method is able to model most types of two phase physical waves. However, if the physical waves are not "dispersive" enough, thus, unavailable to draw power spectrum analysis, the method will not work. Otherwise it should be applicable to most of the cases. Two example of such data are given in the appendix.

Appendix A

The Experiment Data Supplied by Prof. Johnson

The data is provided by Dr. Johnson for the thesis work.
EXOct16_141326_rn_14_w_0.490_o_0.000_g_2.718_bi_0.00_RW_
EXOct16_141326_rn_15_w_0.490_o_0.000_g_2.717_bi_0.00_RW_
EXOct16_141326_rn_39_w_0.294_o_0.000_g_3.538_bi_0.00_RW_
EXOct16_141326_rn_40_w_0.294_o_0.000_g_3.538_bi_0.00_RW_
EXOct16_141326_rn_41_w_0.315_o_0.000_g_3.534_bi_0.00_RW_
EXOct16_141326_rn_42_w_0.335_o_0.000_g_3.531_bi_0.00_RW_
EXOct16_141326_rn_43_w_0.355_o_0.000_g_3.528_bi_0.00_RW_
EXOct16_160117_rn_69_w_0.457_o_0.000_g_3.974_bi_0.00_RW_
EXOct16_160117_rn_70_w_0.474_o_0.000_g_3.972_bi_0.00_RW_
EXOct22_130914_rn_154_w_0.321_o_0.000_g_3.399_bi_0.10_RW_
EXOct22_130914_rn_155_w_0.342_o_0.000_g_3.398_bi_0.10_RW_
EXOct22_130914_rn_156_w_0.362_o_0.000_g_3.393_bi_0.10_RW_
EXOct22_130914_rn_157_w_0.383_o_0.000_g_3.389_bi_0.10_RW_
EXOct22_130914_rn_158_w_0.402_o_0.000_g_3.397_bi_0.10_RW_
EXOct22_130914_rn_159_w_0.421_o_0.000_g_3.394_bi_0.10_RW_
EXOct22_130914_rn_160_w_0.440_o_0.000_g_3.389_bi_0.10_RW_
EXOct22_130914_rn_161_w_0.462_o_0.000_g_3.393_bi_0.10_RW_

Each data consists of six columns of interest. The goal is to model each column. The each column of the experiment measure[1] is listed below in three figures as examples to show their appearances.

[1] EXOct22_130914_rn_161_w_0.462_o_0.000_g_3.393_bi_0.10_RW_

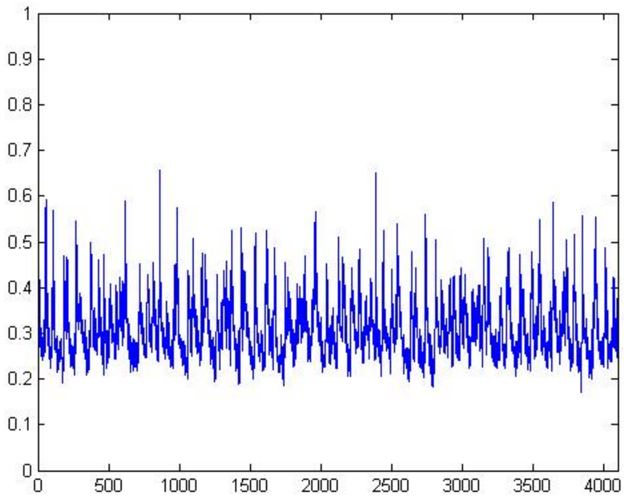


Figure A.1: *The Experiment Measurement of the First Column*

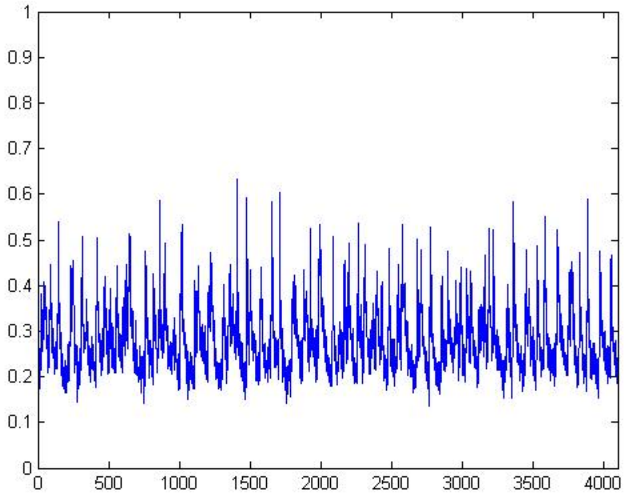


Figure A.2: *The Experiment Measurement of the Second Column*

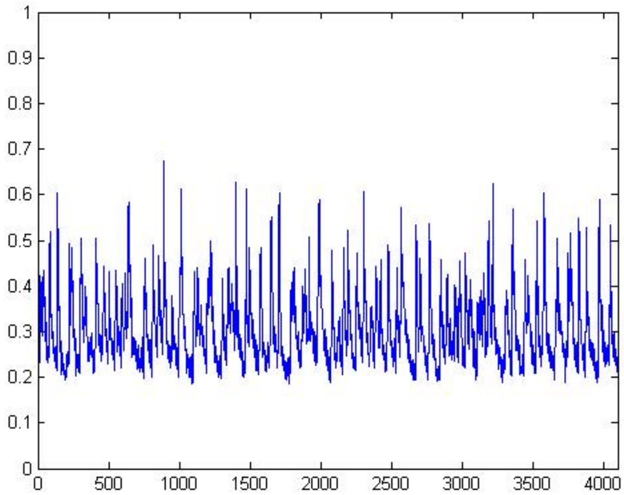


Figure A.3: *The Experiment Measurement of the Sixth Column*

Appendix B

The Two Initial Codes by Prof. Johnson

The codes in original forms first-time I received from Prof. Johnson.

B.0.1 Code for Single Time Trace

```
%clear all
%close all
N=4096;

PN=4;
n=1;

%nBit=1;
%bit = round(rand(1,nBit))
%a_p=(1-2*bit)

%a=-1*(2.0/(2.0)^PN);

WAVES=zeros(1,N+1);
WAVES_CUM=zeros(1,N+1);

HEIGHT=0.20;
TROUGH=0.40;

wl=floor((N+1)*rand);
%wl=0;
k=1;
 for j=wl:N;
     WAVES(k)= (HEIGHT*(j^PN)/(N^PN))+TROUGH;
     k=k+1;
 end
 for j=0:wl-1;
     WAVES(k)= (HEIGHT*(j^PN)/(N^PN))+TROUGH;
     k=k+1;
 end
```

```
%plot(WAVES,'red','LineWidth',2);
%axis([0 N 0 1])
COUNT=1;
WAVE_TRAIN=2;

while WAVE_TRAIN<=2^12;
    AMPLITUDE=(1.0/(WAVE_TRAIN));    % reduces the amplitude of the waves in powers of two
    for ntimes=1:1;
        WAVE_FRAC=zeros(1,N+1);    % entire domain reproduced at each time step
        for m=1:((WAVE_TRAIN));
            %  a_rand=floor(101*rand);
            %while a_rand==50 || a_rand==0 || a_rand==100;
            % while a_rand==50;
            %      a_rand=floor(101*rand);
            % end
            %  if a_rand<50;
            %        a=0;
            %  else
            %        a=1;
            %  end
            %a=-(2.0/(PN+1))*a;

            LOW=fix((m-1)*N/(WAVE_TRAIN));
            HIGH=fix(((m*N)/(WAVE_TRAIN)));
            b=floor((HIGH-LOW+1)*rand+LOW); % introduces a uniformly random starting point
            k=LOW+1;

            for j=b:HIGH;
                WAVE_FRAC(k) = (AMPLITUDE*HEIGHT*((j-LOW)^PN)*(WAVE_TRAIN/N)^PN);
                k=k+1;
            end;
            for j=LOW:b-1;
                WAVE_FRAC(k) = (AMPLITUDE*HEIGHT*((j-LOW)^PN)*(WAVE_TRAIN/N)^PN);
                k=k+1;
            end;
            CREST=AMPLITUDE*HEIGHT;%max(WAVE_FRAC(LOW+1:HIGH));
            TROUGH=0.0;%min(WAVE_FRAC(LOW+1:HIGH));
           % bit = round(rand(1,nBit))
           % a_p=(1-2*bit)
           % a=-1*(2.0/(2.0)^PN);
            if m<WAVE_TRAIN;
                for k=LOW:HIGH-1;
                   % WAVES_CUM(k+1)=(WAVES_CUM(k+1)+ a*(CREST-TROUGH) + WAVE_FRAC(k+1));
                     WAVES_CUM(k+1)=(WAVES_CUM(k+1) - (CREST-TROUGH)/(PN+1) + WAVE_FRAC(k+1));
                end;
                % %                  for k=LOW:HIGH-1;
                %                       if WAVES(k+1)>1;
```

```
                    %                          EXCESS=WAVES(k+1) - 1.0;
                    %                          l=0;
                    %                          while EXCESS>0.0 && (k+1-l)>0;
                    %                              if WAVES(k+1-l)<1.0;
                    %                                  EXCESS_DIFF=(1.0-WAVES(k+1-l))
                    %                                  WAVES(k+1-l)=1.0;
                    %                                  EXCESS=EXCESS-EXCESS_DIFF;
                    %                              end;
                    %                              l=(l+1);
                    %                          end;
                    %                      end;
                    %                  end;
                else
                    for k=LOW:HIGH;
                        WAVES_CUM(k+1)=(WAVES_CUM(k+1) - (CREST-TROUGH)/(PN+1) + WAVE_FRAC(k+1));
                    end;
                end;
                %    end;
            end;
        end;
        COUNT=COUNT+1;
        WAVE_TRAIN=2^COUNT;
end;
for k=1:N;
        WAVES(k+1)=(WAVES(k+1) + WAVES_CUM(k+1));
        if WAVES(k+1)>1;
            EXCESS=WAVES(k+1) - 1.0;
            l=0;
            while EXCESS>0.0 && (k+1-l)>0;
                if WAVES(k+1-l)<1.0;
                    EXCESS_DIFF=(1.0-WAVES(k+1-l))
                    WAVES(k+1-l)=1.0;
                    EXCESS=EXCESS-EXCESS_DIFF;
                end;
                l=(l+1);
            end;
        end;
end;
% for k=LOW:HIGH;
%       WAVES(k+1)=(WAVES(k+1) + WAVES_CUM(k+1));
% end;
%for k in range(0,N):
% print ('%d %.8f'%(k,WAVES[k]))
%end
%hold on
%axis([0 4096 0 1])
%axis([0 N 0 1])
%plot(WAVES,'LineWidth',2);
```

```
%axis([0 N 0 1])
%axis([0 4096 0 1])
%hold off
mean(WAVES)
%WVL=[1:1:N+1];
%WAVES(2,:)=WVL;
%fm=fldim(WAVES)
%mean(WAVES_CUM)
```

B.0.2 The Original Code for Synthetic Wave Train

```
%clear all
%close all
format compact

FL2
VEC_1=WAVES(1,:);
STOP=40;
XX = [1:1:(STOP+1)*(N+1)];

 for i=1:STOP;
     FL2;
     VEC_2=WAVES(1,:);
     VEC_APP=[ VEC_1 VEC_2 ];
     VEC_1=VEC_APP;
 end
EVERY=1;
POINTS=STOP*N/4;

%VEC_APP(2,:)=XX;
%plot(VEC_APP(1,1:10:200),'black','LineWidth',2);
%plot(VEC_APP(1,1:EVERY:(EVERY*POINTS)),'LineWidth',2);
%plot(VEC_APP(1,:))
%hold on
VEC_APP_REV=VEC_APP(end:-1:1)
%plot(VEC_APP_REV(1,1:EVERY:(EVERY*POINTS))*0.8,'red','LineWidth
    ',2);
 plot(VEC_APP_REV(1,1:EVERY:(POINTS))*0.8,'red','LineWidth',2);
 axis([0 N*STOP/100 0 1]);
 plot(VEC_APP)
 axis([0 POINTS 0 1]);
%fldim(VEC_APP)
mean(VEC_APP(1,:))
```

Appendix C

The Corresponding Tables for the Example

1. ***Column One***

EXOct_16_161 Column 1

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

0.1563 -9.6366

0.1140 -7.1957

0.0586 -2.2517

0.0924 -6.9527

0.0741 -5.2522

0.1375 -4.8386

0.1034 -8.5201

0.0765 -5.6412

0.1263 -3.7944

0.1584 -4.5686

0.1407 -3.8239

0.1757 -7.3510

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

0.1540 -3.5020

0.0644 -1.9142

0.1814 -5.4020

0.1320 -5.6953

0.1644 -1.6116

0.0986 -0.2917

0.1630 -3.6116

0.0645 -3.1559

0.1271 -4.5988

0.1457 0.7378

0.1536 -3.0644

0.1355 -1.9580

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

0.0827 -9.3439

0.1233 -9.7725

0.1111 -5.5811

0.1166 -7.3313

0.0522 -6.7537

0.1259 -7.7729

0.1466 -9.6913

0.1699 -6.1877

0.1474 -5.9281

0.0893 -7.7082

0.2308 -9.0399

0.2122 -7.3036

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

0.1223 -14.8219

0.1781 -14.1491

0.0953 -13.7019

0.0707 -11.5080

0.1645 -12.8191

0.0552 -13.9150

0.1395 -15.1592

0.0960 -13.6107

0.0728 -12.2269

0.1503 -11.1701

0.1690 -12.5001

0.1324 -14.6673

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

0.0846 -19.4995

0.0755 -17.6431

0.0379 -19.3851

0.1301 -17.5539

0.1211 -16.2060

0.1254 -19.9994

0.1167 -16.5185

0.0998 -19.9353

0.0571 -18.0894

0.0796 -17.2367

0.1491 -20.7996

0.1038 -19.6181

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

0.1165 -35.4115

0.0577 -34.2731

0.0828 -35.1685

0.1237 -33.8326

0.0935 -35.6451

0.0786 -35.5467

0.1909 -33.9948

0.1987 -33.9166

0.1173 -35.4221

0.1758 -34.9832

0.1146 -34.8267

0.1878 -34.2457

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

2. ***Column Two***

EXOct_16_161 Column 2

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

-0.0249 -9.2505

-0.0721 -7.7517

-0.0935 -6.0977

-0.0237 -7.5622

-0.0471 -6.8354

0.1089 -5.0065

0.0669 -5.9490

0.0071 -6.5169

-0.0307 -6.9191

-0.0546 -4.6288

0.1058 -9.2518

0.0056 -6.5126

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

0.0016 -5.4495

-0.1038 -1.6925

-0.0908 -4.2224

-0.0017 -1.9838

-0.0709 -1.2052

0.0494 -2.1960

-0.0039 -3.3024

-0.0200 -4.2225

-0.0766 -0.7297

-0.0050 -0.7184

-0.0202 -3.7794

-0.0357 -3.2979

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

-0.0258 -8.7710

-0.1073 -7.1162

-0.0624 -7.3026

-0.0595 -6.0024

-0.1472 -6.2319

-0.0633 -5.3345

-0.0962 -6.7215

-0.0696 -5.4683

-0.0255 -6.1892

-0.0570 -6.1422

-0.0701 -9.5603

-0.0404 -7.7738

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

-0.0731 -15.6790

-0.0569 -14.3382

-0.0501 -12.3640

-0.1041 -13.7372

-0.1477 -12.2389

-0.1256 -14.4211

-0.1388 -13.1819

-0.0713 -14.4054

-0.0488 -14.1099

-0.0332 -11.3349

-0.0665 -13.1556

-0.0145 -15.9833

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

-0.1128 -20.6575

-0.1394 -18.5039

-0.1095 -18.7160

-0.1769 -17.6891

-0.1382 -17.8754

-0.0532 -18.0051

-0.1289 -18.1569

-0.1319 -19.0932

-0.0618 -20.0039

-0.0680 -18.4144

-0.1457 -19.1557

-0.0772 -16.9535

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

-0.1597 -37.5749

-0.1349 -35.4660

-0.1712 -34.7824

-0.1766 -35.4689

-0.0697 -34.6252

-0.0712 -33.7847

-0.1480 -35.6984

-0.0738 -35.2191

-0.0732 -35.3371

-0.1027 -33.8708

-0.0756 -35.8174

-0.0670 -36.0662

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

3. ***Column Three***

EXOct_16_161 Column 3

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

-0.0489 -8.8143

-0.1109 -8.1563

-0.0760 -5.1123

-0.0033 -7.4557

0.0415 -7.5623

0.0339 -6.0058

0.0475 -6.0611

-0.0281 -5.8221

-0.0369 -1.1495

-0.1382 -8.7774

-0.0312 -6.9232

-0.1537 -7.3652

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

-0.0598 -2.7885

-0.1376 -3.7330

-0.0869 -2.6944

-0.1112 -5.0393

-0.0825 -1.4303

-0.0677 -2.0312

-0.1744 -4.4176

-0.0848 0.6065

-0.0669 -2.5804

-0.0785 -1.9550

-0.0528 -2.7875

-0.0689 -2.0504

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

-0.0882 -9.8101

-0.1038 -4.4810

-0.1345 -8.4376

-0.0868 -8.7176

-0.0738 -6.0376

-0.0080 -7.4451

-0.0874 -4.7596

-0.0462 -9.0350

-0.0562 -7.1733

-0.1119 -8.6131

-0.1263 -3.8051

-0.0099 -8.0757

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

-0.1317 -15.0785

-0.0513 -13.5301

-0.1024 -12.9660

-0.0118 -12.2600

-0.0720 -12.1645

-0.0605 -12.5875

-0.0460 -12.4126

-0.1074 -13.7276

-0.1266 -14.8255

-0.1204 -13.4755

-0.1077 -12.0995

-0.0767 -13.9752

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

-0.0706 -20.5668

-0.1100 -19.7898

-0.0991 -18.7038

-0.1236 -18.8278

-0.1183 -20.8250

-0.1342 -18.6131

-0.1050 -17.0576

-0.0254 -14.0283

-0.0945 -18.1202

-0.0985 -17.8842

-0.0277 -18.4749

0.0055 -17.5532

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

-0.0747 -34.2855

-0.0359 -36.0329

-0.0907 -35.5372

-0.1238 -35.0730

-0.1507 -34.9207

-0.0749 -35.6364

-0.0419 -34.9060

-0.1289 -36.5009

-0.0189 -35.4279

-0.1254 -35.8616

-0.0679 -34.9778

-0.0394 -35.4585

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

4. ***Column Four***

EXOct_16_161 Column 4

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

-2.3909 -8.8319

-2.3460 -4.9296

-2.3719 -7.1696

-2.4053 -6.8881

-2.3832 -7.2263

-2.3652 -4.6810

-2.3848 -5.5080

-2.3762 -4.4828

-2.4087 -7.5837

-2.3503 -5.4628

-2.3991 -3.3769

-2.3512 -5.1444

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

-1.5350 -3.1852

-1.5281 -3.4950

-1.4613 -3.9943

-1.4428 -6.1364

-1.4506 -1.3846

-1.5623 -1.4800

-1.5537 -0.7722

-1.5609 -5.0232

-1.5488 -1.0108

-1.4723 -2.1182

-1.4983 -2.2510

-1.4290 -7.1017

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

-1.1061 -11.5724

-1.1042 -9.7175

-1.1277 -6.7456

-1.0746 -5.5529

-1.1404 -6.5215

-1.1145 -8.6767

-0.9931 -6.6232

-1.0595 -6.6619

-1.1355 -9.4596

-1.0050 -7.0256

-1.0616 -5.1069

-0.9512 -7.1605

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

-0.8536 -15.5875

-0.8361 -14.6822

-0.7935 -13.3875

-0.8557 -14.0935

-0.7217 -12.7991

-0.8594 -12.5632

-0.7783 -10.4640

-0.8064 -13.0365

-0.7652 -9.8413

-0.8262 -13.0677

-0.7636 -14.7727

-0.8312 -14.1471

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

-0.6531 -20.2187

-0.6668 -16.5174

-0.6226 -18.2909

-0.6081 -19.8066

-0.6813 -18.4698

-0.5927 -16.8184

-0.6025 -17.3494

-0.6800 -19.2844

-0.6163 -18.0123

-0.5249 -17.1425

-0.5797 -16.4208

-0.6252 -19.4995

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

-0.2534 -38.1924

-0.2993 -36.2595

-0.2411 -35.1658

-0.2375 -36.9047

-0.2427 -35.9590

-0.2948 -36.2083

-0.2332 -34.7445

-0.2734 -36.9131

-0.1973 -34.2336

-0.2579 -35.3589

-0.1991 -34.9494

-0.1432 -34.8064

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

5. ***Column Five***

EXOct_16_161 Column 5

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

-2.7082 -6.7508

-2.5639 -4.9573

-2.6250 -7.3137

-2.7158 -6.6774

-2.6396 -5.6598

-2.6904 -8.3539

-2.6975 -6.4260

-2.7037 -7.3311

-2.6431 -5.1649

-2.6032 -5.1221

-2.6642 -4.8117

-2.6316 -4.5121

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

-1.6602 -4.1410

-1.6560 -3.2808

-1.7001 -1.0720

-1.6994 -3.6390

-1.6506 -2.2980

-1.6553 -0.7140

-1.7697 -2.6718

-1.7213 -3.7201

-1.6547 -2.1471

-1.6576 -3.5643

-1.6675 -3.1520

-1.6257 -3.7734

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

-1.2442 -11.3346

-1.2715 -9.2461

-1.2228 -7.7343

-1.2400 -5.8663

-1.2065 -7.8307

-1.1840 -6.8261

-1.1497 -6.8146

-1.0932 -5.9560

-1.0993 -8.0397

-1.2201 -3.2045

-1.1936 -7.3305

-1.1987 -7.0651

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

-0.9647 -15.8969

-0.9331 -12.3019

-0.9538 -13.1974

-0.9232 -11.4299

-0.8833 -14.7603

-0.9407 -14.5042

-0.8014 -14.1592

-0.8658 -10.4672

-0.9456 -10.6652

-0.9465 -12.1367

-0.9319 -13.9758

-0.8806 -14.7813

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

-0.7199 -21.4894

-0.7654 -18.8577

-0.7278 -19.8201

-0.7807 -18.5621

-0.7200 -17.5868

-0.7256 -18.6735

-0.6765 -17.8600

-0.6331 -14.9684

-0.6845 -19.1137

-0.6615 -16.8167

-0.6842 -17.5680

-0.6532 -16.6552

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

-0.2078 -39.1409

-0.2649 -35.5978

-0.2562 -34.8762

-0.3052 -34.8482

-0.3199 -36.0660

-0.3089 -34.3255

-0.2602 -34.4628

-0.2444 -35.9725

-0.1614 -33.6923

-0.1962 -33.4719

-0.2494 -35.8913

-0.1968 -35.6960

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

6. ***Column Six***

EXOct_16_161 Column 6

Percentage Error Between Synthetic Waves and Experiment Measurement

Polynomial Degree of 1 with NG from 1 to 12

MEAN STD

-0.0341 -7.5630

0.0363 -6.2864

0.0550 -7.2470

0.0502 -6.7335

-0.0507 -4.4099

0.0287 -6.5148

0.0962 -5.8587

-0.0257 -6.4329

0.0898 -8.3421

-0.0054 -6.2714

0.0694 -4.0470

-0.0067 -7.9388

Polynomial Degree of 2 with NG from 1 to 12

MEAN STD

-0.0453 -3.2197

0.0320 -3.8735

-0.0041 -3.6171

-0.0081 -3.0540

0.0247 0.1075

0.0076 -3.3138

0.0221 -2.0871

0.0561 -2.3103

-0.0153 -0.7506

0.0144 -3.8105

0.0988 0.0041

-0.0193 -0.6561

Polynomial Degree of 3 with NG from 1 to 12

MEAN STD

0.0155 -11.9360

0.0459 -7.0205

0.0446 -7.8252

0.0443 -8.4797

0.0132 -7.2356

0.0038 -7.4055

-0.0011 -7.6483

-0.0008 -7.5273

0.1030 -9.0860

0.1082 -7.5897

-0.0077 -9.1289

0.1003 -6.2629

Polynomial Degree of 4 with NG from 1 to 12

MEAN STD

0.0281 -15.1736

-0.0009 -13.1845

0.0026 -12.4463

0.0504 -13.0715

0.1113 -10.4854

0.0022 -13.5435

0.0649 -14.0821

0.0879 -13.7684

-0.0047 -15.5030

0.0609 -10.3178

0.0899 -13.7393

0.0982 -13.8510

Polynomial Degree of 5 with NG from 1 to 12

MEAN STD

-0.0268 -19.4728

0.1075 -19.2282

-0.0338 -15.6506

0.0793 -14.6425

0.0515 -18.8549

-0.0539 -17.6258

0.0090 -17.1459

0.0583 -16.9304

0.0671 -18.5977

0.0873 -16.3967

0.0004 -17.7949

0.1450 -18.8549

Polynomial Degree of 10 with NG from 1 to 12

MEAN STD

-0.0426 -37.7198

-0.0441 -33.1298

0.0217 -34.5641

-0.0219 -35.5186

0.0652 -34.1389

-0.0080 -35.0112

0.0640 -35.9756

0.0404	-35.2855
0.0315	-34.1552
0.0298	-34.6860
0.0541	-34.9914
-0.0006	-36.6241

Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations

Appendix D

Program Code

D.1 MATLAB Code

The FL2 and FL_LOOP are the codes provided by Dr. Johnson, which could be enable me start work efficiently.

D.1.1 ox.m the main calling function

```
function [jj]=ox(x,k,z,cc,dn,cn,l,m) % x=H_omega, k,cc=COLUMN, z=Nbox, dn=Set No.
% cn=Column No, l=mean value, m=standard deviation

[VC_AP]=my(x,k,z,cc,dn,cn);
jj=per(VC_AP,l,m);

filename=['EXOct_16_' num2str(dn) ' Column ' num2str(cn)];
fileID = fopen(filename,'w');
fprintf(fileID,filename);
fprintf(fileID,' \r\n');
fprintf(fileID,'Percentage Error Between Synthetic Waves and Experiment Measurement\r\n');
fprintf(fileID,' \r\n');

m=0;
for i=[1 2 3 4 5 10]

    string1=['Polynomial Degree of ' num2str(i) ' with NG from 1 to 12\r\n'];
    fprintf(fileID,' \r\n');
    fprintf(fileID,string1);
    fprintf(fileID,'%12s %24s\r\n','MEAN','STD');
    %fprintf(fileID,'%12.4f %24.4f\r\n',jj);
    vk=[jj((m+1):(m+12),1)';jj((m+1):(m+12),2)'];
    fprintf(fileID,'%12.4f %24.4f\r\n',vk);
    m=m+12;

end
```

```
fprintf(fileID,' \r\n');
fprintf(fileID,'Notation: STD=standard deviation, MEAN=mean value, NG=No. of Generations');



fclose(fileID);

end
```

D.1.2 Fl2.m the single synthetic time trace

```
%clear all
%close all
N=4096;
PN=t;
n=1;

WAVES=zeros(1,N+1);
WAVES_CUM=zeros(1,N+1);

HEIGHT=x;
TROUGH=y;

wl=floor((N+1)*rand);
%wl=0;
k=1;
 for j=wl:N;
     WAVES(k)= (HEIGHT*(j^PN)/(N^PN))+TROUGH;
     k=k+1;
 end
 for j=0:wl-1;
     WAVES(k)= (HEIGHT*(j^PN)/(N^PN))+TROUGH;
     k=k+1;
 end

COUNT=1;
WAVE_TRAIN=2;

while WAVE_TRAIN<=2^3; % 5-->w
    AMPLITUDE=(1.0/(WAVE_TRAIN));    % reduces the amplitude of the waves in powers of two
    for ntimes=1:1;
        WAVE_FRAC=zeros(1,N+1);    % entire domain reproduced at each time step
        for m=1:((WAVE_TRAIN));


            LOW=fix((m-1)*N/(WAVE_TRAIN));
            HIGH=fix(((m*N)/(WAVE_TRAIN)));
```

```
            b=floor((HIGH-LOW+1)*rand+LOW); % introduces a uniformly random starting point
            k=LOW+1;

            for j=b:HIGH;
                WAVE_FRAC(k) = (AMPLITUDE*HEIGHT*((j-LOW)^PN)*(WAVE_TRAIN/N)^PN);
                k=k+1;
            end;
            for j=LOW:b-1;
                WAVE_FRAC(k) = (AMPLITUDE*HEIGHT*((j-LOW)^PN)*(WAVE_TRAIN/N)^PN);
                k=k+1;
            end;
            CREST=AMPLITUDE*HEIGHT;%max(WAVE_FRAC(LOW+1:HIGH));
            TROUGH=0.0;%min(WAVE_FRAC(LOW+1:HIGH));

            if m<WAVE_TRAIN;
                for k=LOW:HIGH-1;
                    % WAVES_CUM(k+1)=(WAVES_CUM(k+1)+ a*(CREST-TROUGH) + WAVE_FRAC(k+1));
                     WAVES_CUM(k+1)=(WAVES_CUM(k+1) - (CREST-TROUGH)/(PN+1) + WAVE_FRAC(k+1));
                end;

            else
                for k=LOW:HIGH;
                    WAVES_CUM(k+1)=(WAVES_CUM(k+1) - (CREST-TROUGH)/(PN+1) + WAVE_FRAC(k+1));
                end;
            end;
            %    end;
        end;
    end;
    COUNT=COUNT+1;
    WAVE_TRAIN=2^COUNT;
end;
for k=1:N;
    WAVES(k+1)=(WAVES(k+1) + WAVES_CUM(k+1));
    if WAVES(k+1)>1;
        EXCESS=WAVES(k+1)-1.0;
        l=0;
        while EXCESS>0.0 && (k+1-l)>0;
            if WAVES(k+1-l)<1.0;
                EXCESS_DIFF=(1.0-WAVES(k+1-l))
                WAVES(k+1-l)=1.0;
                EXCESS=EXCESS-EXCESS_DIFF;
            end;
            l=(l+1);
        end;
    end;
end;

mean(WAVES);
```

D.1.3 my.m the figure generation function

```
% x=HEIGHT, y=TROUGH, z=Number of Boxes, t=Degree of Polynomials, w=Number
% of Generations, k=Column from the experiment
function [TT]=my (x,k,z,cc,dn,cn)
VC_1=zeros(1,0);
VCC_1=zeros(1,0);

zz=TR(k); % TROUGH Calculation
l=1;
for t=[1 2 3 4 5 10] % t-values are the polynomial degrees, in which is related to the calculations of TROUGH.
    y=zz(l); % l represents the position No. of the element in the vector t
    l=l+1;

    % OX, TR function must be altered when changing t in this code to
    % accommodate the degree of polynomials.
    % OX-- the txt file output part
    % TR-- the calculation of TROUGH

 for w=1:12 % w are the No. of generations
        FL_LOOP;
        TEST=VEC_APP_REV(1:(z+1):length(VEC_APP_REV)); % Synthetic Time Traces
        % Plot Synthetic Time Traces and Experiment Together, Exp->red
        % color

        subplot (2,1,1);
        plot(TEST);
        hold on
        plot (cc,'red');
        hold off
        string1=['Time\_Traces\_Exp (red)\_' num2str(dn) '\_Col' num2str(cn) ' AND Time\_Traces\_SYN (blue)\_' num2str(dn) '\_Col\_' num2str(cn) ];
        string2=['Time\_Traces\_EXP\_' num2str(dn) ' & SYN for Poly\_Degree\_' num2str(t) ' & with No. of Generations\_' num2str(w)];
        xlabel (string1)
        ylabel('Normalized Liquid Height');
        title(string2);

        axis ([0 1500 0 1])
        drawnow

    %synfilename=['Syn_PD' num2str(t) ' & NG ' num2str(w)];
```

```
      %saveas (gcf, synfilename, 'jpg')

      % Mean & STD
      Mn=mean (TEST);
      Sd=std (TEST);
      [AA]=[Mn Sd];
      VC_2=[AA];
      VC_AP=[VC_1;VC_2];
      VC_1=VC_AP;


    % PD & Freq
    % Combined figure with PD & PS

    freq=[0:299]/82;
    [p]=PowerPlot (TEST);

    subplot (2,1,2);
    semilogx(freq,p);
    hold on

    [p]=PowerPlot(cc);

    semilogx(freq,p,'red');
    hold off
    string3=['Power\_EXP\_SYN\_' num2str(dn) '\_Col\_' num2str(cn)];
    string4=['Power\_Spectrum\_Poly Degree\_' num2str(t) ' of EXP\_' num2str(dn) ' & SYN for No. of Generations\_' num2str(w)];
    xlabel (string3);
    ylabel ('Power Density')
    title(string4);
    drawnow


    %PSfilename=['PS_PD ' num2str(t) ' NG ' num2str(w)];
    %saveas (gcf, PSfilename, 'jpg')

    %%subplot file name

    subfilename=['EXP_SYN_' num2str(dn) ' Time Traces_Poly Degree_' num2str(t) '_NG_' num2str(w)];
    saveas (gcf,subfilename, 'jpg')

    [ax]=maxnd (p,2);
    VCC_2=[ax];
    VCC_AP=[VCC_1; VCC_2];
    VCC_1=VCC_AP;

    % Matrix Combination
    TT=[VC_AP VCC_AP];
end

```

```
end
 end
```

D.1.4 FL_LOOP.m function for wave train.

```
%clear all
%close all
format compact

FL2;
VEC_1=WAVES(1,:);
%STOP=z;
STOP=42;
XX = [1:1:(STOP+1)*(N+1)];

 for i=1:STOP;
     FL2;
     VEC_2=WAVES(1,:);
     VEC_APP=[ VEC_1 VEC_2 ];
     VEC_1=VEC_APP;
end
EVERY=1;
POINTS=STOP*N/4;

%VEC_APP(2,:)=XX;
%plot(VEC_APP(1,1:10:200),'black','LineWidth',2);
%plot(VEC_APP(1,1:EVERY:(EVERY*POINTS)),'LineWidth',2);
%plot(VEC_APP(1,:))
%hold on
VEC_APP_REV=VEC_APP(end:-1:1);
%plot(VEC_APP_REV(1,1:EVERY:(EVERY*POINTS))*0.8,'red','LineWidth
    ',2);
%plot(VEC_APP_REV(1,1:EVERY:(POINTS))*0.8,'red','LineWidth',2);
%axis([0 N*STOP/100 0 1]);
%plot(VEC_APP)
%axis([0 POINTS 0 1]);
%fldim(VEC_APP)
%mean(VEC_APP(1,:))
```

D.1.5 Other functions

The following functions are used for computing percentage, trough, power spectrum, the 2nd largest value on the spectra etc..

```
% Percentage
function [cc]=per (VC_AP,av,std)


% fomula given by Dr.Johnson is Error=100(T-E)/E, T->theoretical
    ,
```

```
% E->experiment
a=VC_AP(:,1);
b=VC_AP(:,2);
c_1=100*(a-av)*(1/av);
c_2=100*(b-std)*(1/std);
cc=[c_1  c_2];
end

% Calculating the trough value
function  [TB]=TR (x) % x is the scaled input experiment data
     H_exp=mean (x);
     H_w=2*sqrt(2)*std(x);

     n=1;
     p1=H_exp-H_w./(n+1);
     n=2;
     p2=H_exp-H_w./(n+1);
     n=3;
     p3=H_exp-H_w./(n+1);
     n=4;
     p4=H_exp-H_w./(n+1);  % TROUGH at PD=4
     n=5;
     p5=H_exp-H_w./(n+1);
     n=10;
     p10=H_exp-H_w./(n+1);
     [TB]=[p1 p2 p3 p4 p5 p10];
    end

% The Power Spectrum
function p=PowerPlot (x)
     N=600;
     T=82;
     t=[0:N-1]/N;
     t=t*T;
     p=abs(fft(x)/(N-2));
     p=p(1:N/2).^2;
     %freq=[0:N/2-1]/T;
     %semilogx(freq,p);
end

% Power Density & Corresponding Frequency Display

function y= maxnd(x,n)


    [xu,ind] = unique(x);
    y = [xu(end-n+1) ind(end-n+1)];
    p=xu(end-n+1);
    z=ind (end-n+1);
    N=600;
    T=82;
    freq=[0:N/2-1]/T;
    f=freq (z);

```

```
    y=[p, f]; % Spectrum vector
end
```

Appendix E

Table of Experiment Measurements Statistics

The following tables are obtained from analyzing the experimental measurements. The table will be essential to obtain the comprehensive output table and figures, in which we can then choose the better matched ones arbitrarily.

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3683	0.0611	0.1728	0.8399	0.1818	0.0041	0.5000	41.0000
0.3354	0.0670	0.1895	0.7377	0.1861	0.0039	0.5000	41.0000
0.3477	0.0672	0.1901	0.7742	0.1886	0.0027	0.5000	41.0000
0.3620	0.0688	0.1946	0.7463	0.1848	0.0028	0.3293	27.0000
0.3449	0.0661	0.1870	0.7373	0.2036	0.0052	0.5000	41.0000
0.3470	0.0731	0.2068	0.7218	0.2065	0.0055	0.5000	41.0000

Figure E.1: EXOct16_141326_rn_14_w_0.490_o_0.000_g_2.718_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3689	0.0640	0.1810	0.7629	0.2121	0.0066	0.4756	39.0000
0.3362	0.0708	0.2003	0.7483	0.1543	0.0069	0.4756	39.0000
0.3476	0.0686	0.1940	0.7944	0.1775	0.0062	0.4756	39.0000
0.3622	0.0713	0.2017	0.7628	0.2121	0.0048	0.4756	39.0000
0.3445	0.0678	0.1918	0.6722	0.1919	0.0088	0.4756	39.0000
0.3466	0.0767	0.2169	0.7886	0.2071	0.0097	0.4756	39.0000

Figure E.2: EXOct16_141326_rn_15_w_0.490_o_0.000_g_2.717_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2807	0.0407	0.1151	0.4708	0.1712	0.0013	0.6220	51.0000
0.2623	0.0451	0.1276	0.5137	0.1274	0.0014	0.6220	51.0000
0.2755	0.0473	0.1338	0.4958	0.0935	0.0016	0.4390	36.0000
0.2917	0.0456	0.1290	0.5072	0.1369	0.0014	0.7317	60.0000
0.2717	0.0391	0.1106	0.4137	0.1592	0.0015	0.6220	51.0000
0.2717	0.0431	0.1219	0.5119	0.1504	0.0017	0.6220	51.0000

Figure E.3: EXOct16_141326_rn_39_w_0.294_o_0.000_g_3.538_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2822	0.0386	0.1092	0.4381	0.1634	0.0007	0.8415	69.0000
0.2508	0.0437	0.1236	0.4528	0.1173	0.0008	1.1585	95.0000
0.2653	0.0441	0.1247	0.4864	0.1161	0.0009	0.4512	37.0000
0.2817	0.0438	0.1239	0.4837	0.1198	0.0009	0.4512	37.0000
0.2546	0.0364	0.1030	0.4556	0.1568	0.0009	0.3780	31.0000
0.2542	0.0399	0.1129	0.4184	0.1575	0.0009	0.4512	37.0000

Figure E.4: EXOct16_141326_rn_40_w_0.294_o_0.000_g_3.538_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2863	0.0457	0.1293	0.4945	0.1719	0.0013	0.4634	38.0000
0.2535	0.0525	0.1485	0.5550	0.1462	0.0015	0.5000	41.0000
0.2681	0.0534	0.1510	0.5576	0.1231	0.0020	0.5000	41.0000
0.2838	0.0521	0.1474	0.5432	0.1105	0.0017	0.4512	37.0000
0.2602	0.0451	0.1276	0.4756	0.1633	0.0017	0.4512	37.0000
0.2594	0.0504	0.1426	0.6005	0.1603	0.0013	0.5366	44.0000

Figure E.5: EXOct16_141326_rn_41_w_0.315_o_0.000_g_3.534_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2917	0.0463	0.1310	0.5043	0.1681	0.0014	0.7683	63.0000
0.2599	0.0510	0.1442	0.5473	0.1400	0.0016	0.6951	57.0000
0.2740	0.0514	0.1454	0.5116	0.1358	0.0019	0.6951	57.0000
0.2901	0.0529	0.1496	0.6101	0.1441	0.0022	0.5976	49.0000
0.2668	0.0469	0.1327	0.4673	0.1469	0.0022	0.7683	63.0000
0.2676	0.0506	0.1431	0.5808	0.1548	0.0018	0.7195	59.0000

Figure E.6: EXOct16_141326_rn_42_w_0.335_o_0.000_g_3.531_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2951	0.0496	0.1403	0.5541	0.1773	0.0012	0.5976	49.0000
0.2623	0.0567	0.1604	0.5791	0.1400	0.0016	0.5976	49.0000
0.2755	0.0572	0.1618	0.5531	0.1036	0.0014	0.4878	40.0000
0.2917	0.0584	0.1652	0.6921	0.1376	0.0022	0.5976	49.0000
0.2717	0.0501	0.1417	0.5739	0.1625	0.0016	0.5976	49.0000
0.2717	0.0562	0.1590	0.6065	0.1683	0.0017	0.5976	49.0000

Figure E.7: EXOct16_141326_rn_43_w_0.355_o_0.000_g_3.528_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2913	0.0631	0.1785	0.6109	0.1475	0.0045	0.6707	55.0000
0.2608	0.0693	0.1960	0.6302	0.1187	0.0035	0.6707	55.0000
0.2746	0.0694	0.1963	0.6443	0.0793	0.0032	0.6707	55.0000
0.2890	0.0688	0.1964	0.5818	0.1273	0.0028	0.9878	81.0000
0.2770	0.0613	0.1734	0.5393	0.1519	0.0050	0.6707	55.0000
0.2795	0.0677	0.1915	0.5925	0.1553	0.0043	0.6707	55.0000

Figure E.8: EXOct16_160117_rn_69_w_0.457_o_0.000_g_3.974_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2942	0.0649	0.1836	0.6221	0.1547	0.0027	0.6829	56.0000
0.2635	0.0707	0.2000	0.6735	0.1099	0.0034	0.6951	57.0000
0.2770	0.0700	0.1980	0.6621	0.0935	0.0029	0.6951	57.0000
0.2917	0.0708	0.2003	0.6703	0.1017	0.0034	0.6951	57.0000
0.2818	0.0645	0.1824	0.5982	0.1502	0.0028	0.6463	53.0000
0.2840	0.0690	0.1952	0.5899	0.1487	0.0039	0.6289	56.0000

Figure E.9: EXOct16_160117_rn_70_w_0.474_o_0.000_g_3.972_bi_0.00_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2908	0.0458	0.1295	0.4933	0.1393	0.0017	0.4512	37.0000
0.2594	0.0523	0.1479	0.5829	0.1281	0.0018	0.4512	37.0000
0.2762	0.0529	0.1496	0.5250	0.1366	0.0018	0.3049	25.0000
0.2905	0.0543	0.1536	0.6236	0.1469	0.0021	0.4756	39.0000
0.2687	0.0470	0.1329	0.4999	0.1502	0.0019	0.4512	37.0000
0.2697	0.0513	0.1451	0.5406	0.1694	0.0021	0.4512	37.0000

Figure E.10: EXOct22_130914_rn_154_w_0.321_o_0.000_g_3.399_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2952	0.0482	0.1363	0.5518	0.1727	0.0019	0.7805	64.0000
0.2654	0.0553	0.1564	0.5283	0.1421	0.0023	0.7805	64.0000
0.2799	0.0565	0.1598	0.5454	0.0480	0.0019	0.7805	64.0000
0.2942	0.0549	0.1553	0.6033	0.1640	0.0020	0.5976	49.0000
0.2750	0.0500	0.1414	0.5038	0.1689	0.0020	0.7805	64.0000
0.2762	0.0553	0.1564	0.5315	0.1710	0.0024	0.5122	42.0000

Figure E.11: EXOct22_130914_rn_155_w_0.342_o_0.000_g_3.398_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.2976	0.0527	0.1491	0.6804	0.1681	0.0026	0.4268	35.0000
0.2654	0.0605	0.1711	0.6357	0.1421	0.0024	0.7683	63.0000
0.2811	0.0607	0.1717	0.6392	0.1144	0.0027	0.7683	63.0000
0.2952	0.0597	0.1689	0.6298	0.1376	0.0031	0.7683	63.0000
0.2798	0.0553	0.1564	0.5569	0.1568	0.0030	0.4268	35.0000
0.2797	0.0620	0.1754	0.6749	0.1763	0.0029	0.7683	63.0000

Figure E.12: EXOct22_130914_rn_156_w_0.362_o_0.000_g_3.393_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3020	0.0533	0.1508	0.5736	0.1742	0.0033	0.6341	52.0000
0.2728	0.0605	0.1711	0.5824	0.1441	0.0033	0.6341	52.0000
0.2874	0.0597	0.1689	0.6340	0.1440	0.0026	0.2927	24.0000
0.3003	0.0603	0.1706	0.6851	0.1182	0.0024	0.2927	24.0000
0.2854	0.0568	0.1607	0.5506	0.1681	0.0043	0.6341	52.0000
0.2876	0.0628	0.1776	0.6555	0.1731	0.0045	0.6341	52.0000

Figure E.13: EXOct22_130914_rn_157_w_0.383_o_0.000_g_3.389_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3033	0.0558	0.1578	0.6014	0.1877	0.0023	0.7805	64.0000
0.2746	0.0641	0.1813	0.6302	0.1151	0.0045	0.7805	64.0000
0.2892	0.0632	0.1788	0.6830	0.0972	0.0032	0.7805	64.0000
0.3020	0.0616	0.1742	0.6642	0.1081	0.0023	0.7805	64.0000
0.2892	0.0591	0.1672	0.6454	0.1785	0.0026	0.7805	64.0000
0.2922	0.0671	0.1898	0.6526	0.1800	0.0050	0.7805	64.0000

Figure E.14: EXOct22_130914_rn_158_w_0.402_o_0.000_g_3.397_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3056	0.0602	0.1703	0.6505	0.1825	0.0048	0.7561	62.0000
0.2752	0.0690	0.1952	0.7468	0.2752	0.0046	0.7561	62.0000
0.2907	0.0669	0.1892	0.6551	0.1544	0.0036	0.7561	62.0000
0.3037	0.0699	0.1977	0.6565	0.1427	0.0033	0.7561	62.0000
0.2929	0.0631	0.1785	0.6454	0.1777	0.0056	0.7561	62.0000
0.2942	0.0715	0.2022	0.6674	0.1790	0.0063	0.7561	62.0000

Figure E.15: EXOct22_130914_rn_159_w_0.421_o_0.000_g_3.394_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3078	0.0603	0.1706	0.6576	0.1810	0.0042	0.5610	46.0000
0.2787	0.0687	0.1943	0.7007	0.1421	0.0041	0.4512	37.0000
0.2938	0.0684	0.1935	0.7044	0.1231	0.0037	0.4512	37.0000
0.3071	0.0669	0.1892	0.6725	0.1405	0.0034	0.4512	37.0000
0.2968	0.0636	0.1799	0.5948	0.1617	0.0051	0.5610	46.0000
0.3002	0.0722	0.2042	0.6872	0.1731	0.0053	0.4512	37.0000

Figure E.16: EXOct22_130914_rn_160_w_0.440_o_0.000_g_3.389_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3106	0.0647	0.1830	0.6576	0.1719	0.0054	0.5122	42.0000
0.2802	0.0701	0.1983	0.6329	0.1372	0.0082	0.5122	42.0000
0.2971	0.0698	0.1974	0.6030	0.1239	0.0081	0.5122	42.0000
0.3105	0.0703	0.1988	0.6763	0.1553	0.0083	0.5122	42.0000
0.3012	0.0685	0.1937	0.6842	0.1657	0.0053	0.5122	42.0000
0.3038	0.0752	0.2127	0.6734	0.1858	0.0100	0.5122	42.0000

Figure E.17: EXOct22_130914_rn_161_w_0.462_o_0.000_g_3.393_bi_0.10_RW_

Mean	STD	H_omega	Max	Min	MP	Freq	Nbox
0.3151	0.0666	0.1884	0.7227	0.1727	0.0033	0.5366	44.0000
0.2860	0.0744	0.2104	0.8025	0.1288	0.0047	0.7683	63.0000
0.3025	0.0733	0.2073	0.7237	0.1117	0.0047	0.4146	34.0000
0.3025	0.0733	0.2073	0.7237	0.1117	0.0047	0.4146	34.0000
0.3093	0.0702	0.1986	0.6308	0.1560	0.0039	0.5976	49.0000
0.3123	0.0779	0.2203	0.6607	0.1619	0.0054	0.7683	63.0000

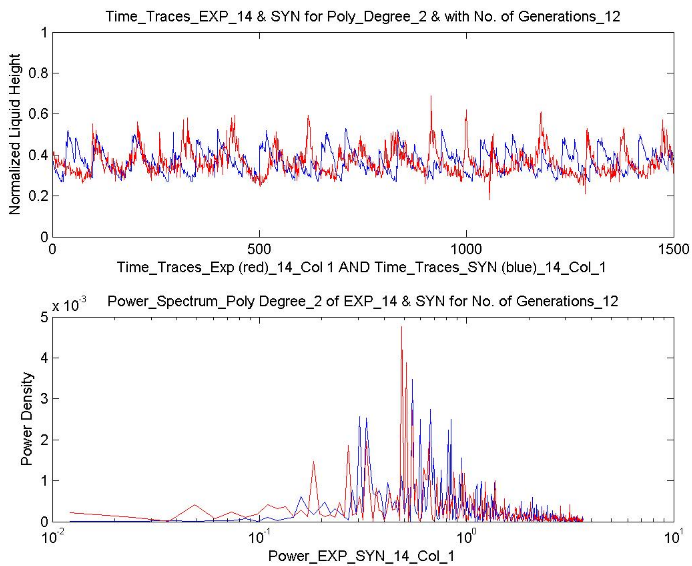
Figure E.18: EXOct22_130914_rn_163_w_0.499_o_0.000_g_3.391_bi_0.10_RW_

Appendix F

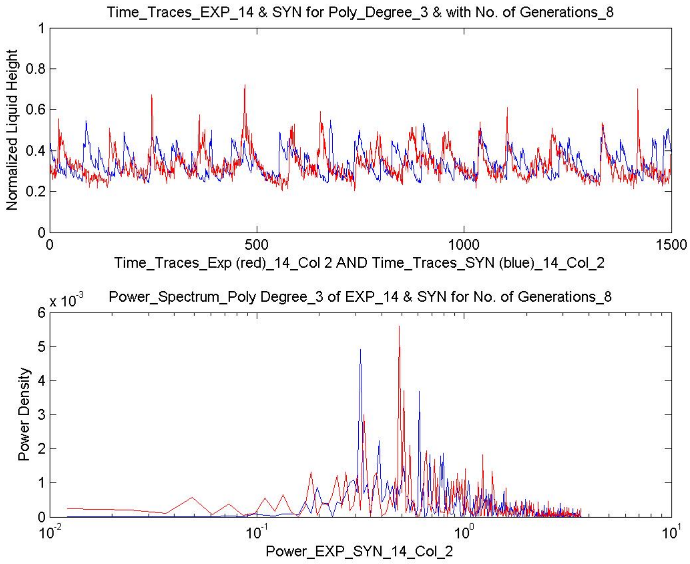
List of Figures of Good Matches to Experiment Measurements

1. **EXOct16_141326_rn_14_w_0.490_o_0.000_g_2.718_bi_0.00_RW_**

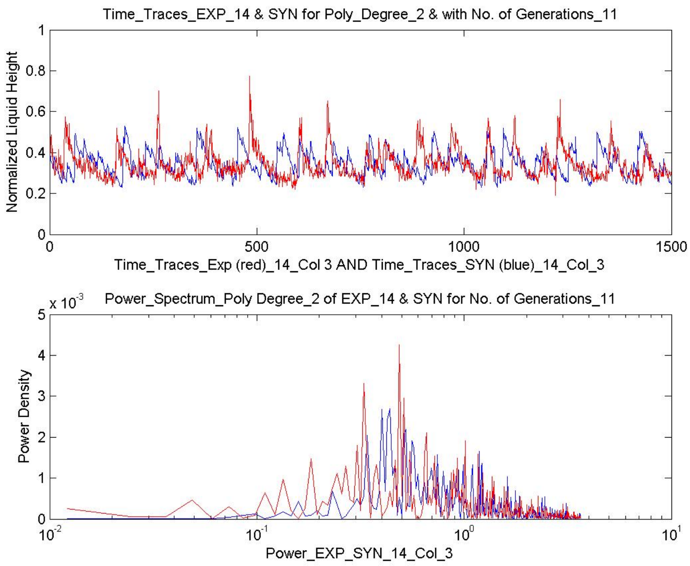
   (a) ***Column One***



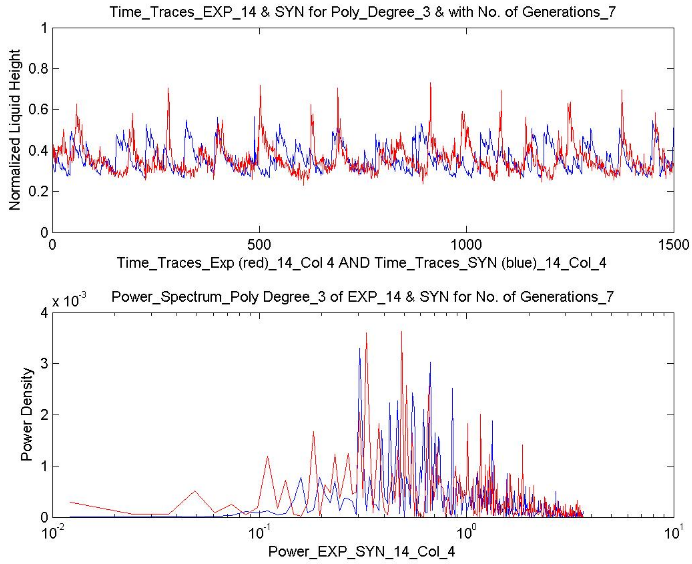
   (b) ***Column Two***



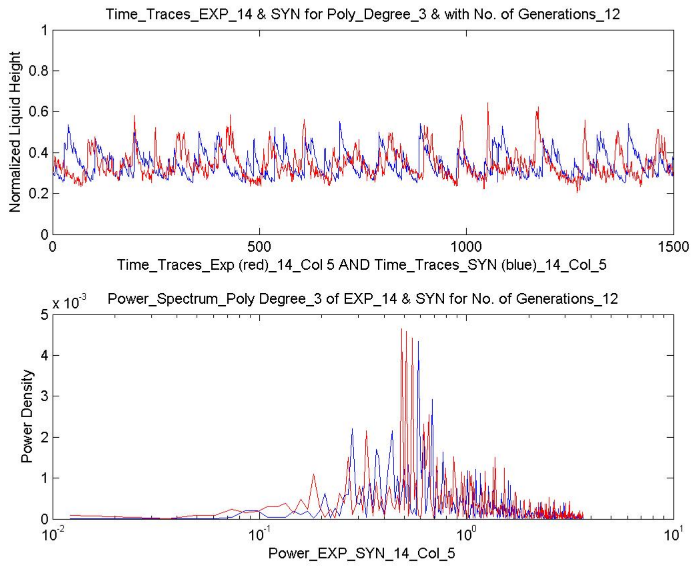
(c) ***Column Three***



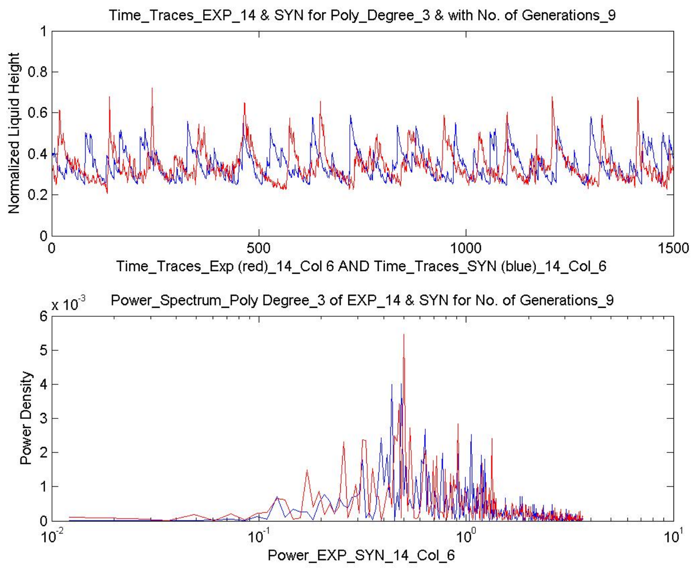
(d) ***Column Four***



(e) ***Column Five***

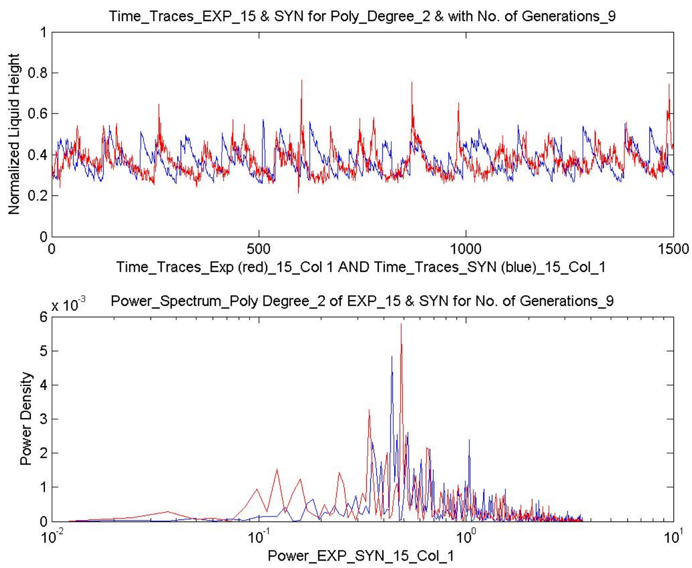


(f) ***Column Six***

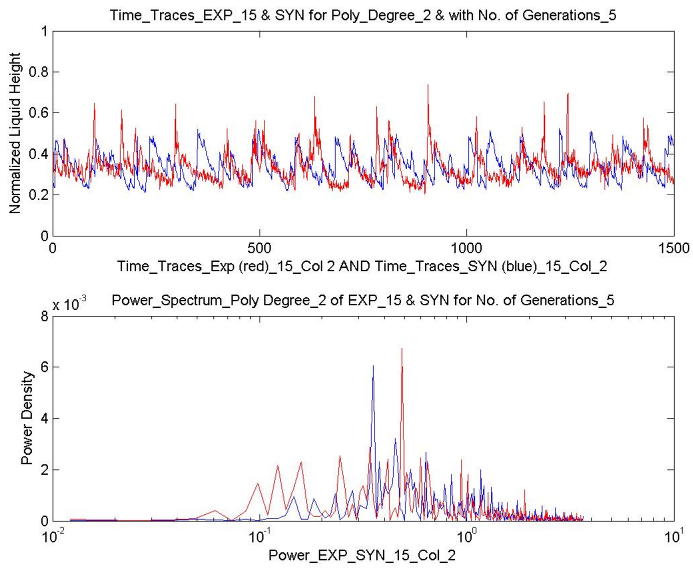


2. **EXOct16_141326_rn_15_w_0.490_o_0.000_g_2.717_bi_0.00_RW_**

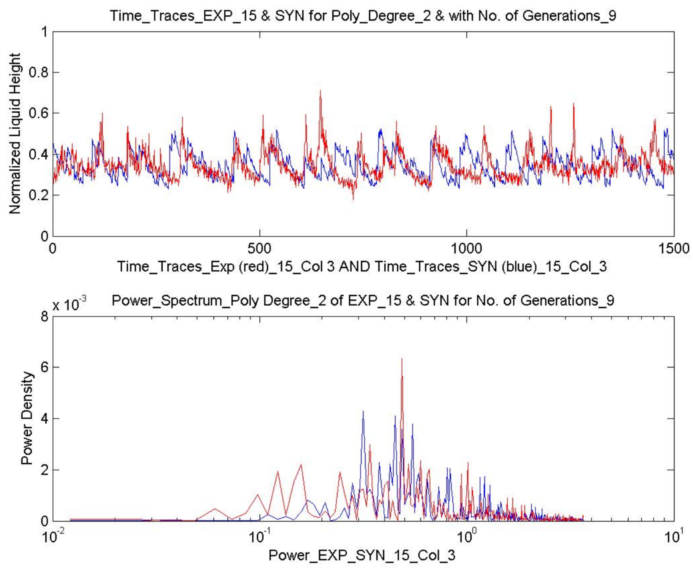
   (a) ***Column One***



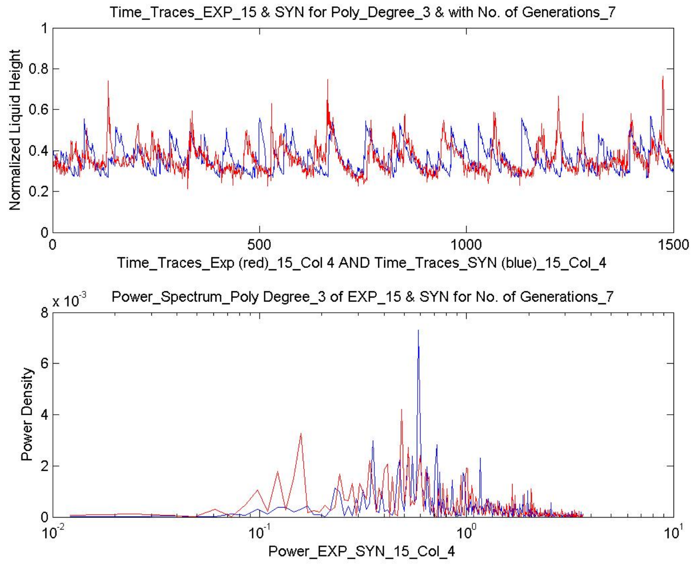
   (b) ***Column Two***



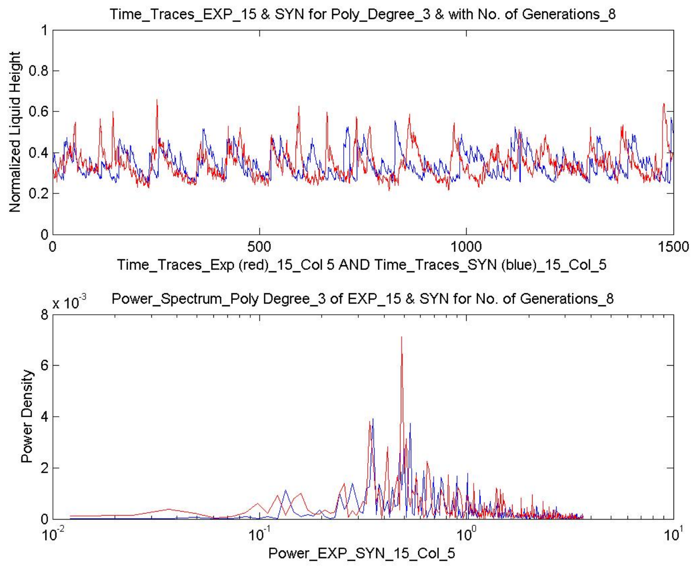
(c) ***Column Three***



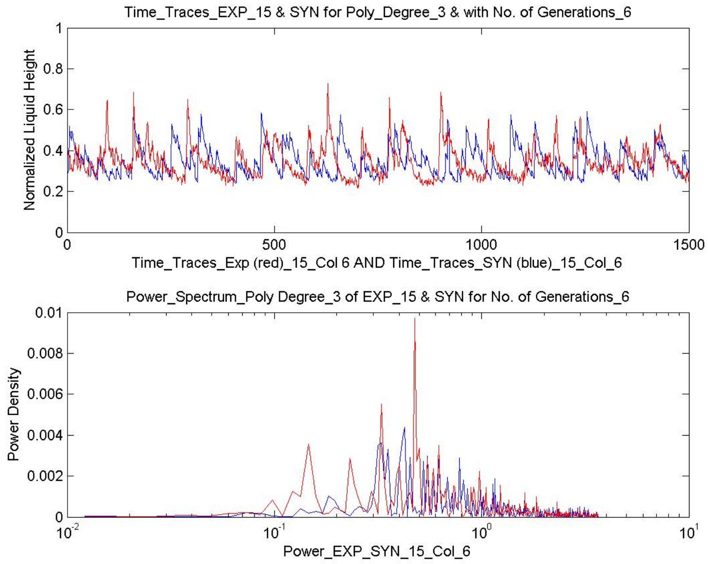
(d) ***Column Four***



(e) ***Column Five***

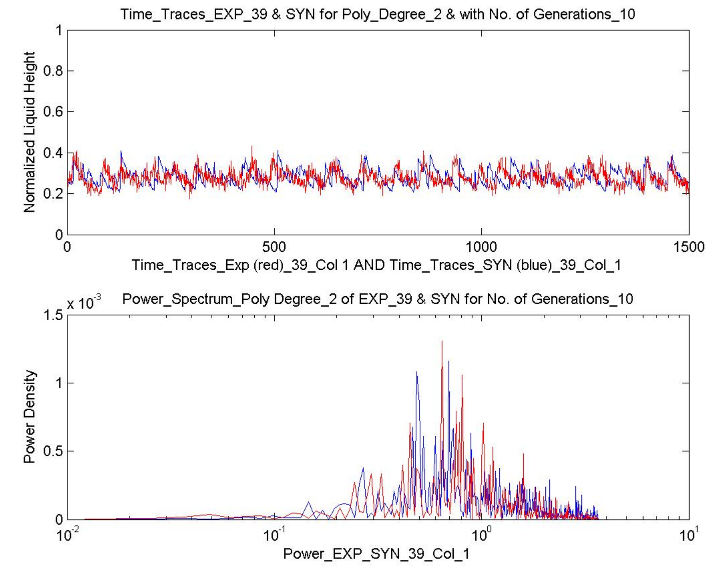


(f) ***Column Six***

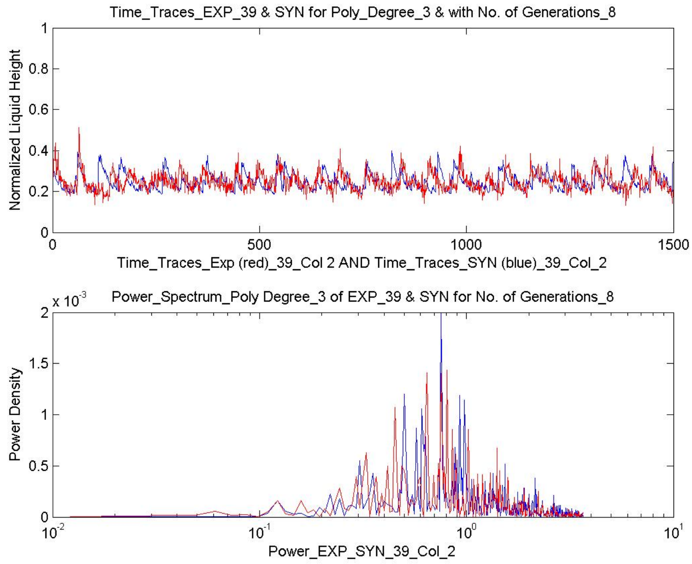


3. **EXOct16_141326_rn_39_w_0.294_o_0.000_g_3.538_bi_0.00_RW_**

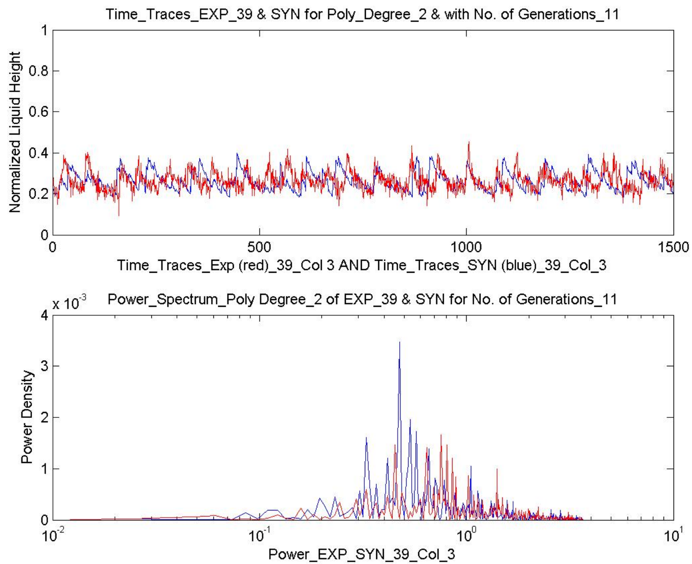
   (a) ***Column One***



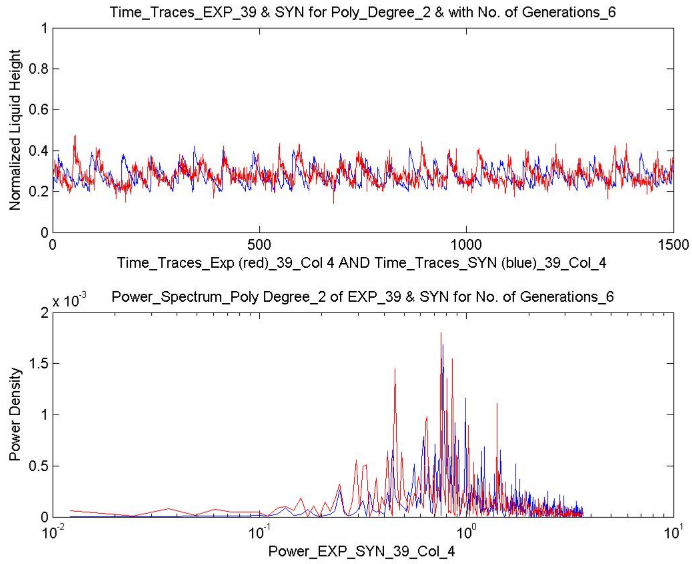
   (b) ***Column Two***



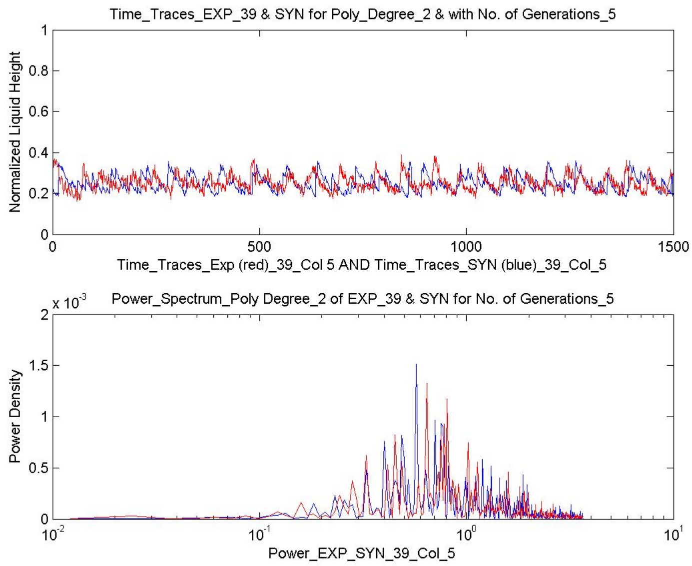
(c) ***Column Three***



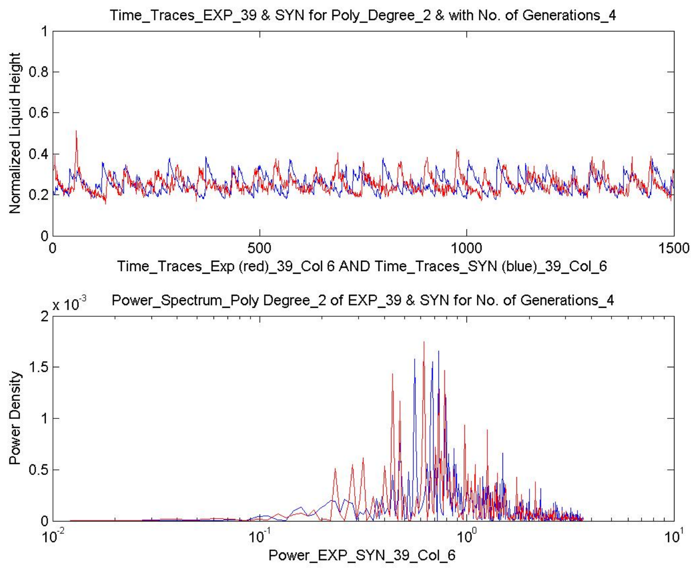
(d) ***Column Four***



(e) ***Column Five***

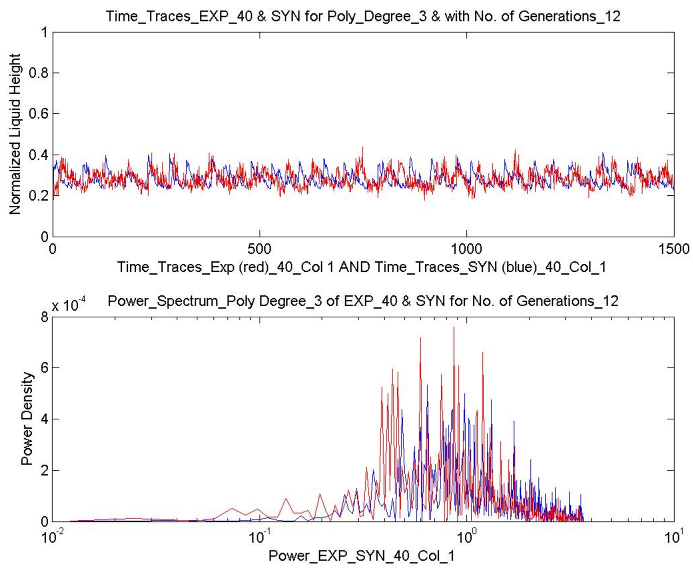


(f) ***Column Six***

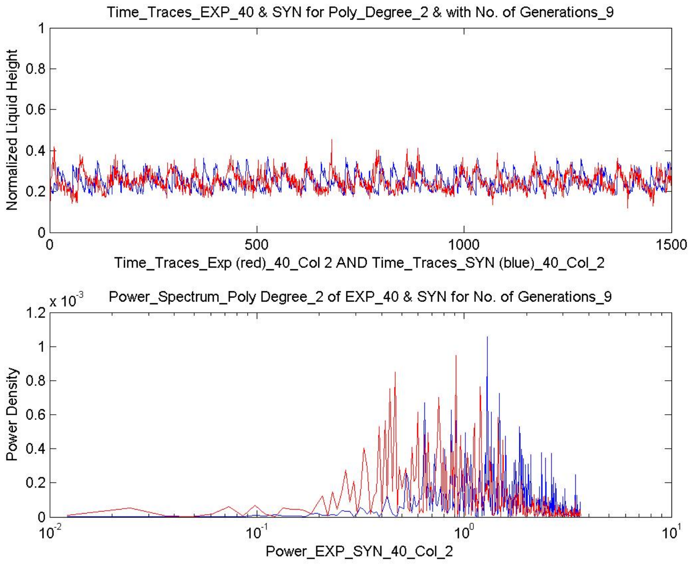


4. **EXOct16_141326_rn_40_w_0.294_o_0.000_g_3.538_bi_0.00_RW_**

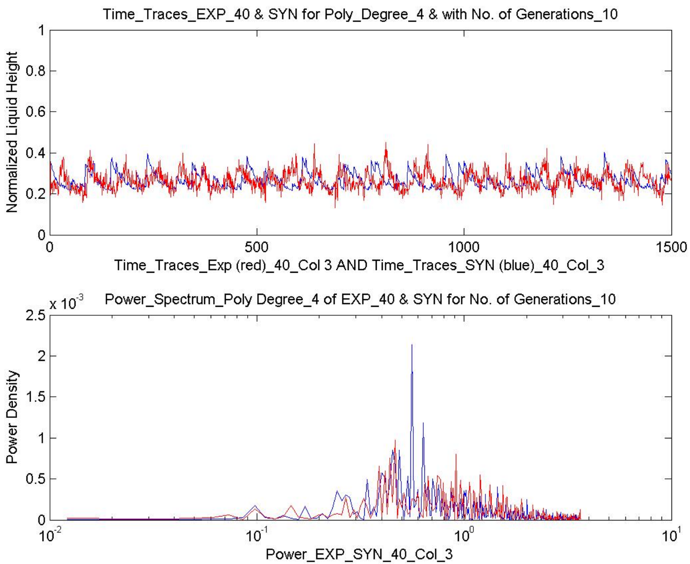
   (a) ***Column One***



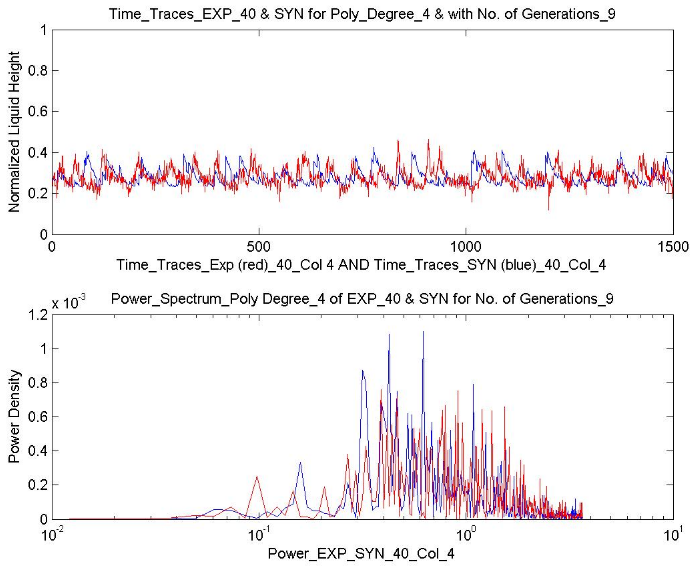
   (b) ***Column Two***



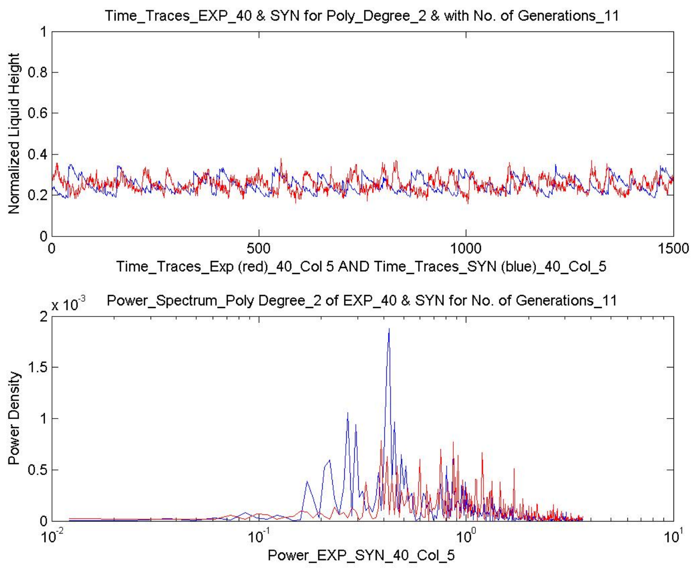
(c) ***Column Three***



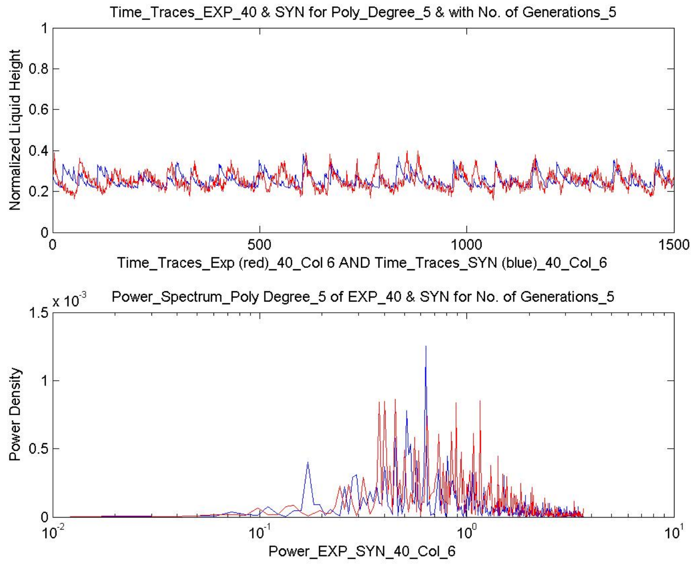
(d) ***Column Four***



(e) ***Column Five***

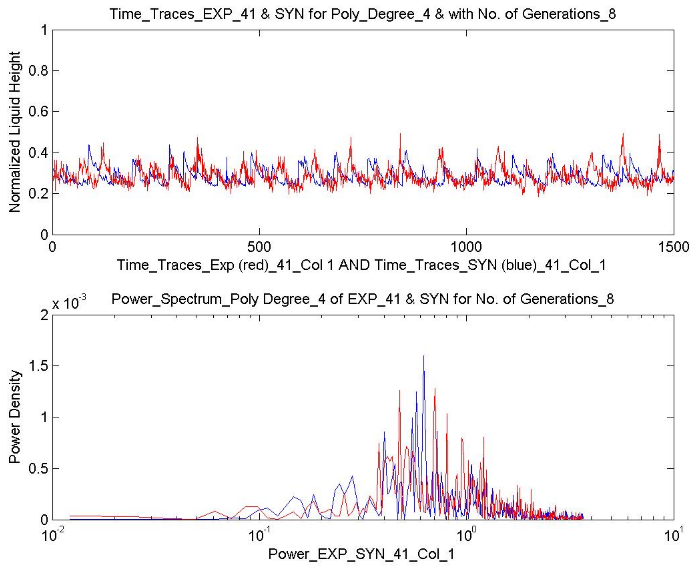


Column Six

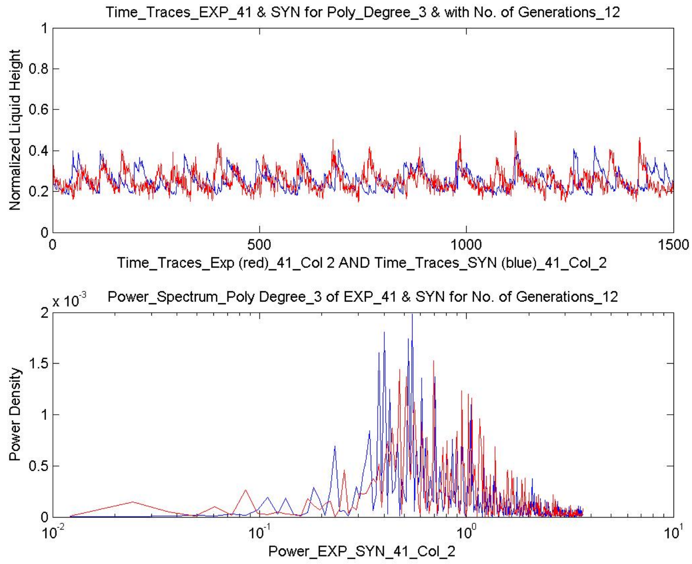


5. **EXOct16_141326_rn_41_w_0.315_o_0.000_g_3.534_bi_0.00_RW_**

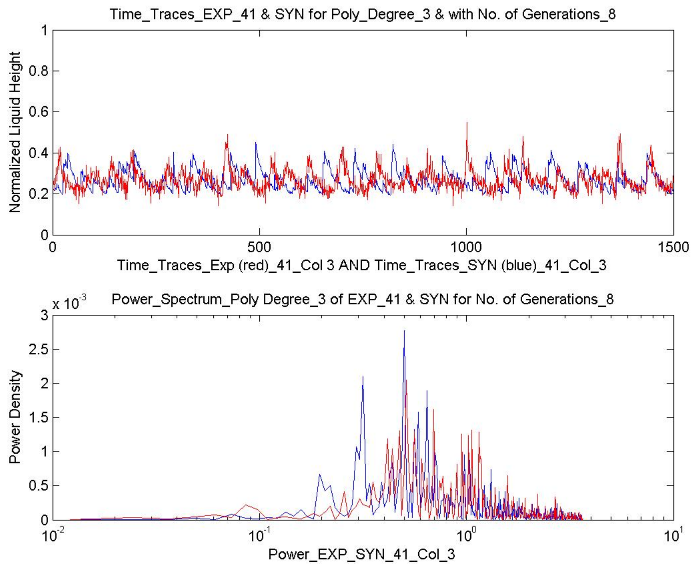
   (a) ***Column One***



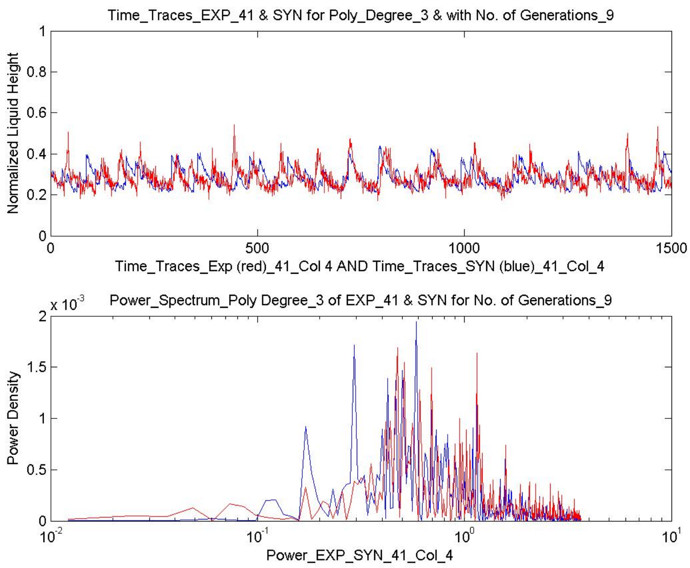
   (b) ***Column Two***



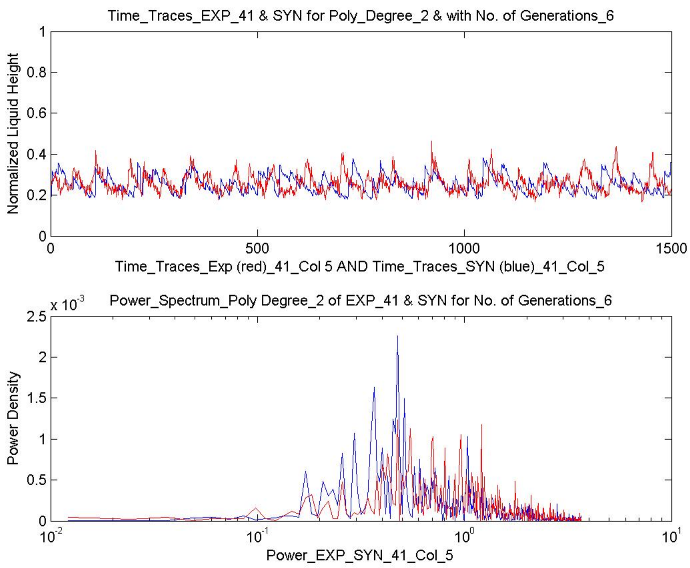
(c) ***Column Three***



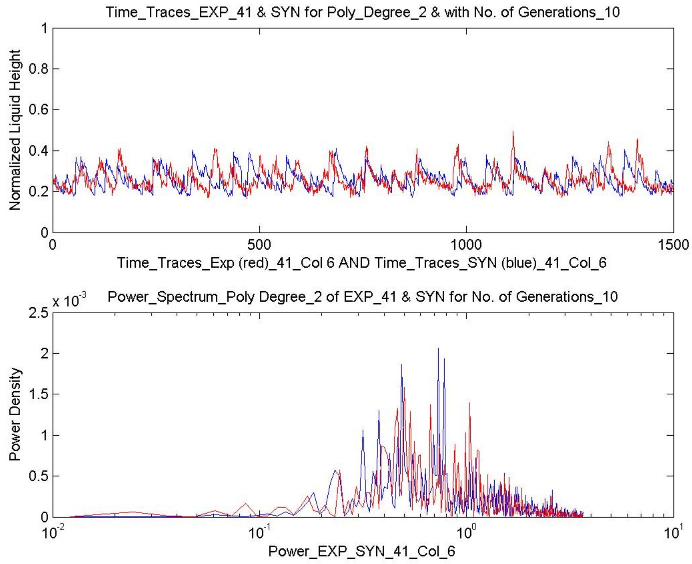
(d) ***Column Four***



(e) ***Column Five***

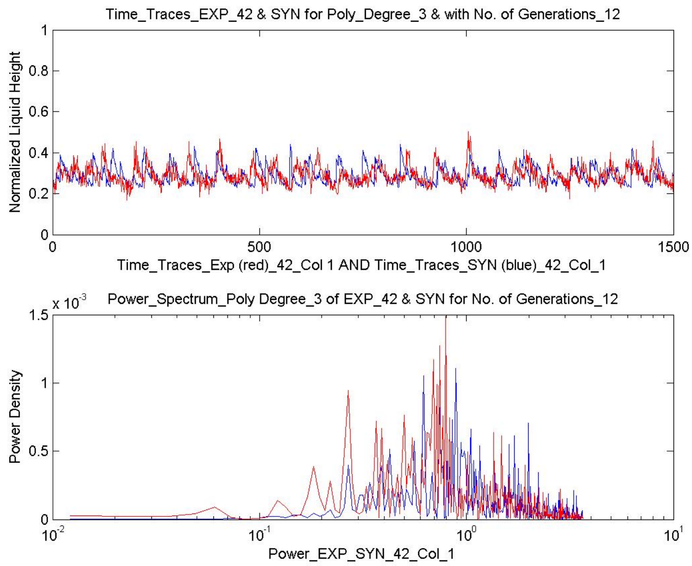


(f) ***Column Six***

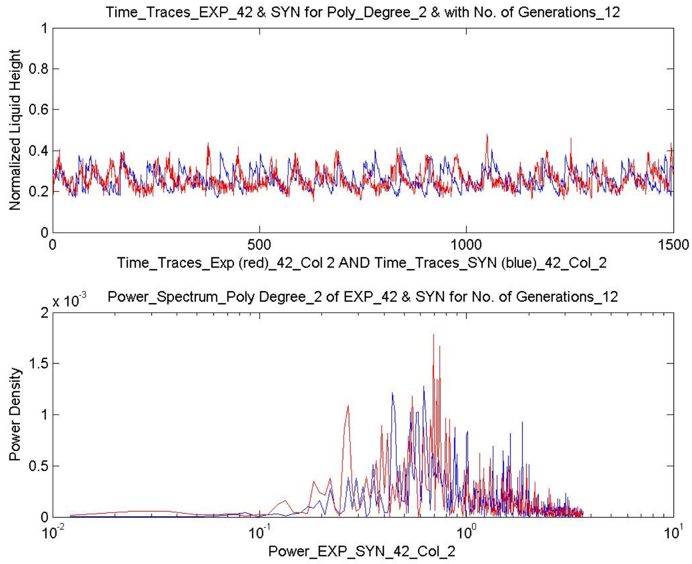


6. **EXOct16_141326_rn_42_w_0.335_o_0.000_g_3.531_bi_0.00_RW_**

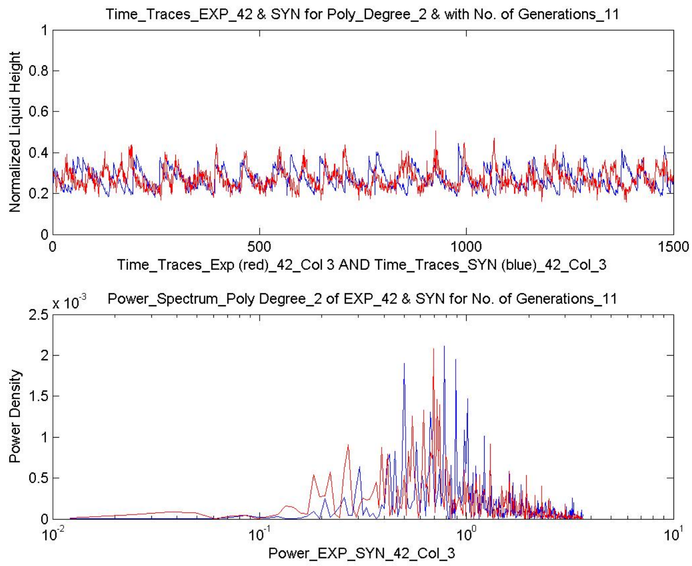
   (a) ***Column One***



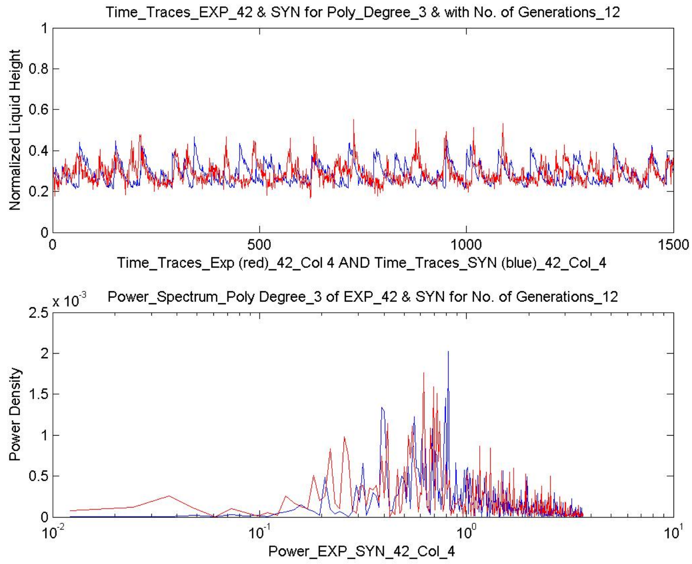
   (b) ***Column Two***



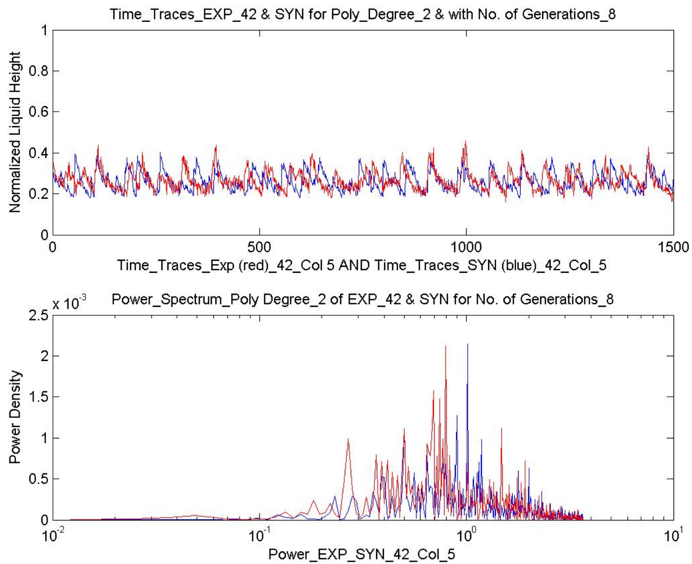
(c) ***Column Three***



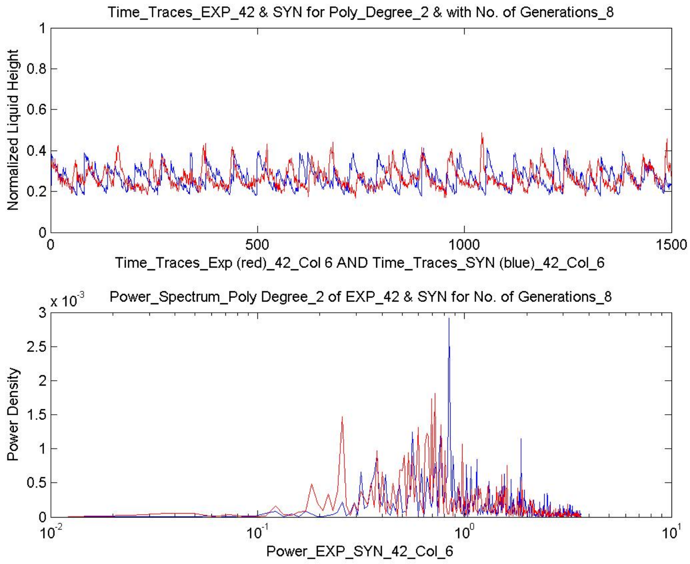
(d) ***Column Four***



(e) ***Column Five***

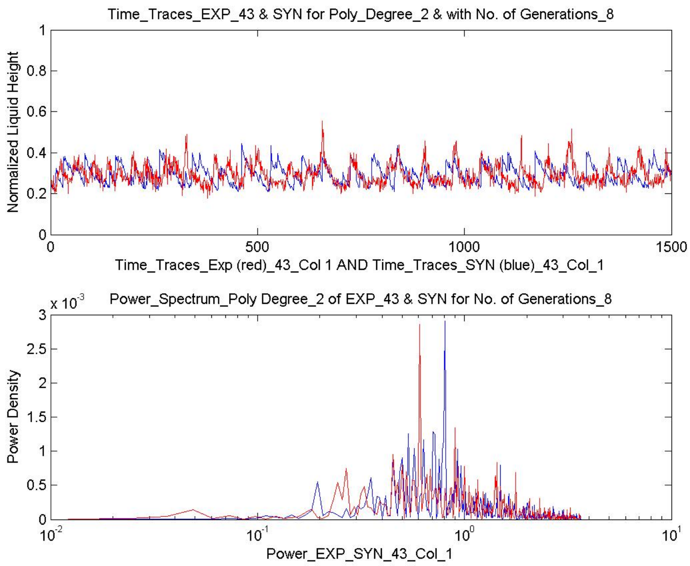


(f) ***Column Six***

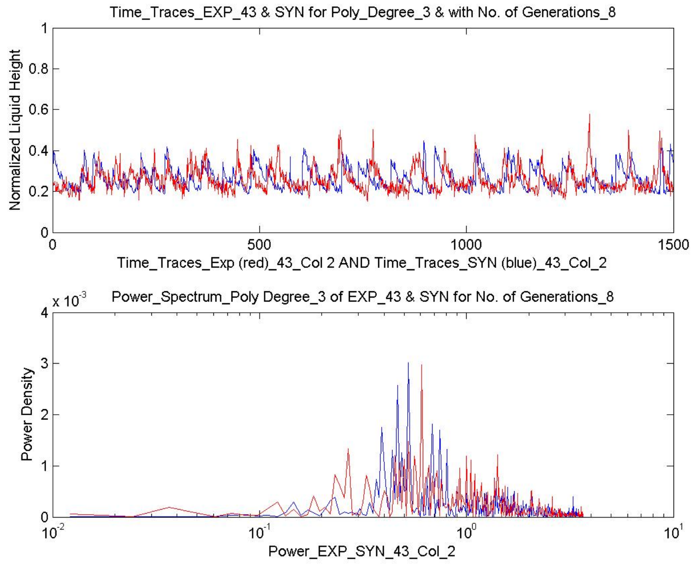


7. **EXOct16_141326_rn_43_w_0.355_o_0.000_g_3.528_bi_0.00_RW_**

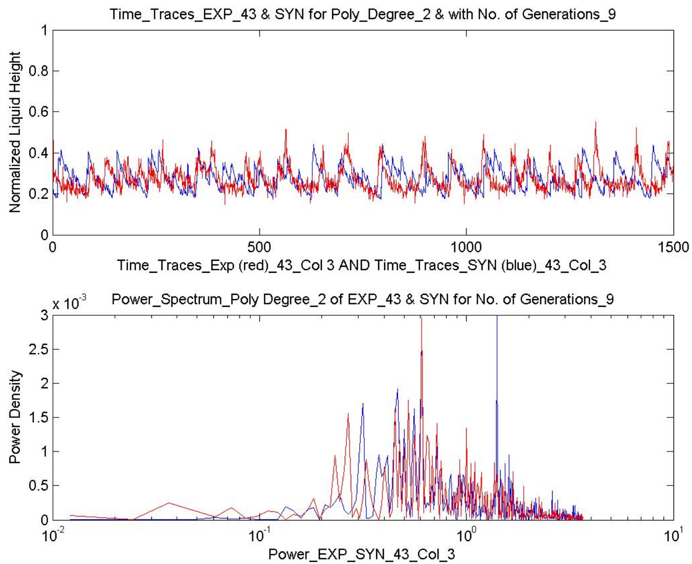
   (a) ***Column One***



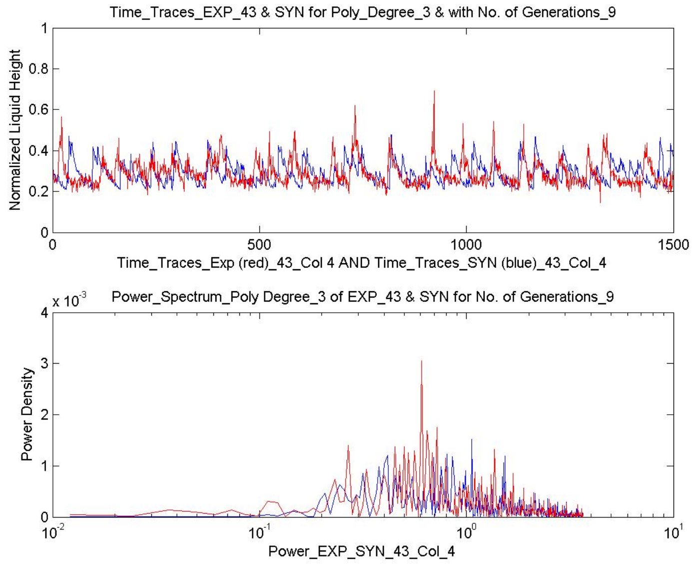
   (b) ***Column Two***



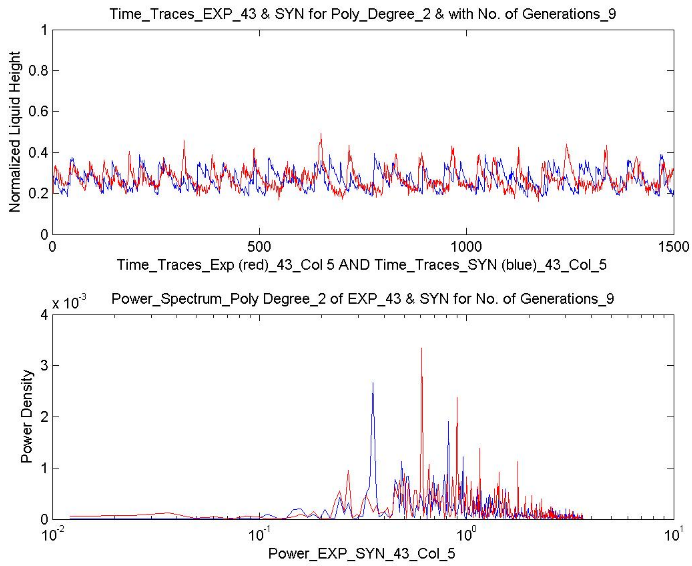
(c) ***Column Three***



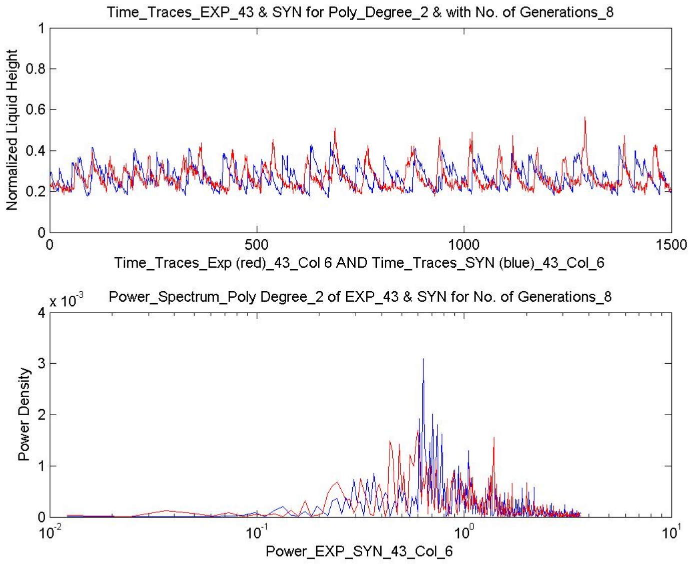
(d) ***Column Four***



(e) ***Column Five***

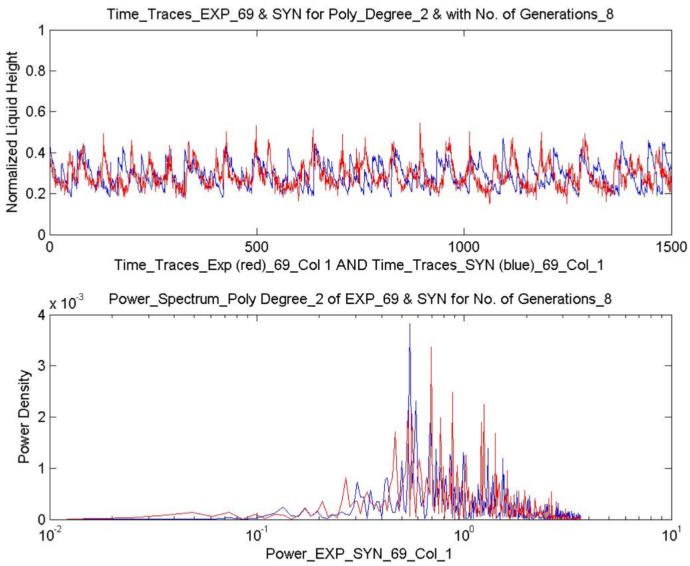


(f) ***Column Six***

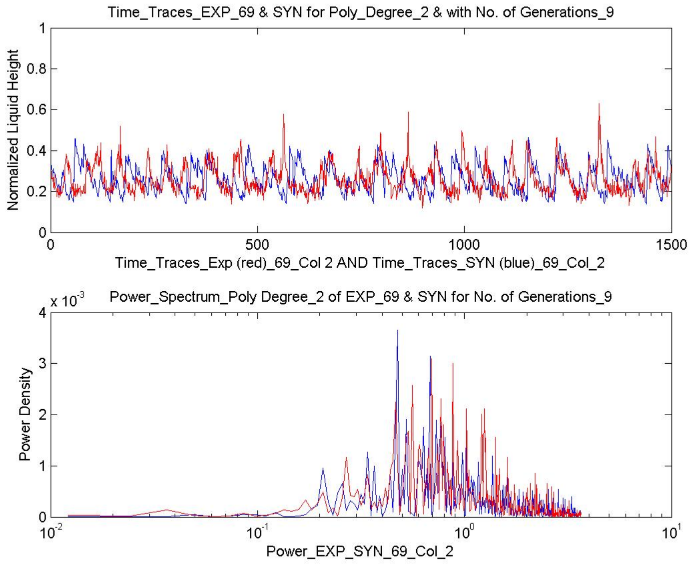


8. **EXOct16_160117_rn_69_w_0.457_o_0.000_g_3.974_bi_0.00_RW_**

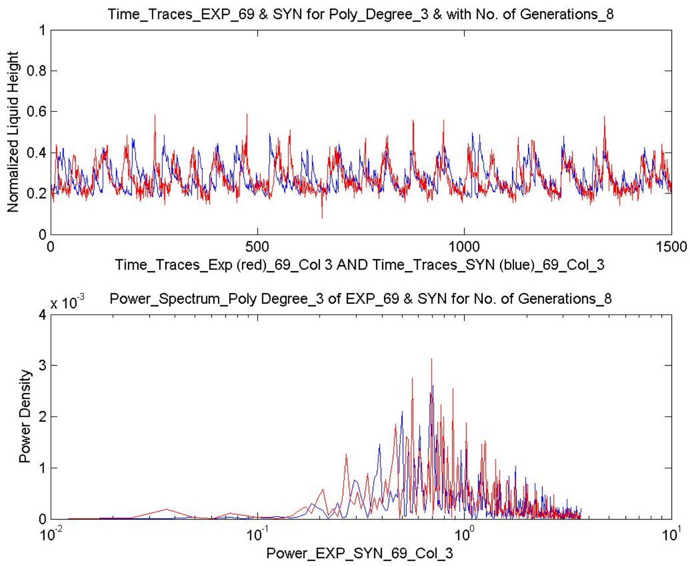
   (a) ***Column One***



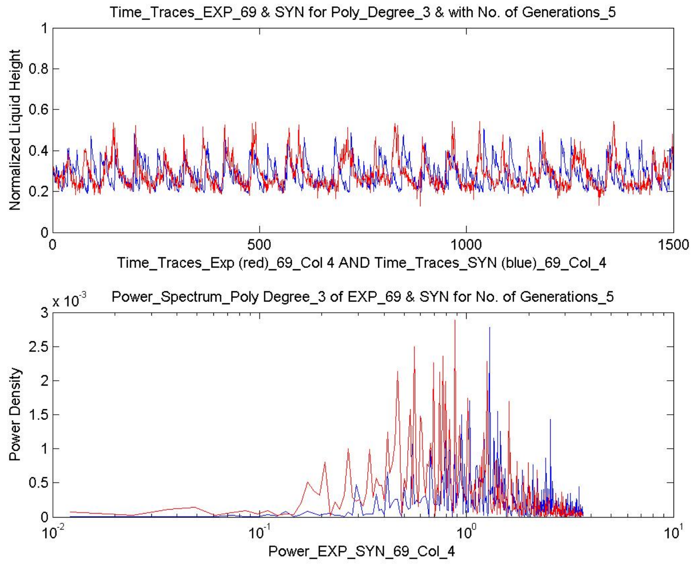
   (b) ***Column Two***



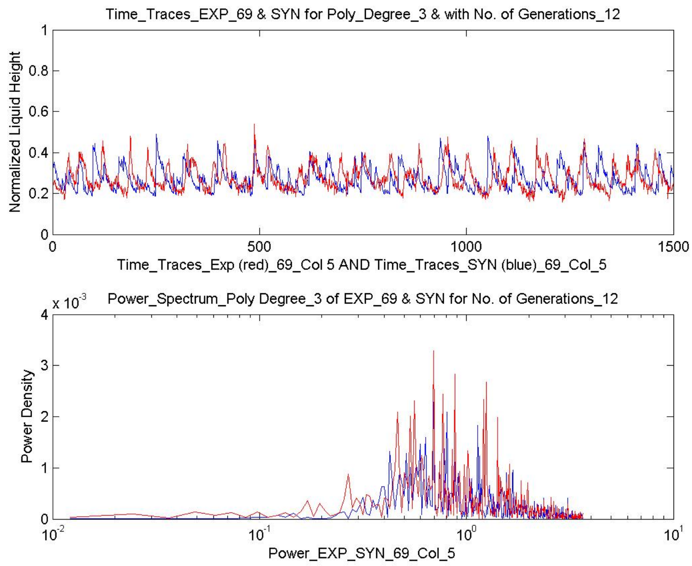
(c) ***Column Three***



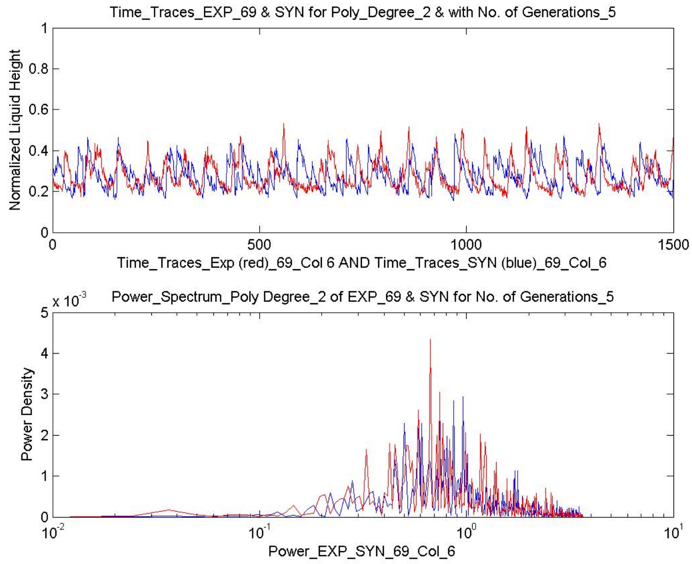
(d) ***Column Four***



(e) ***Column Five***

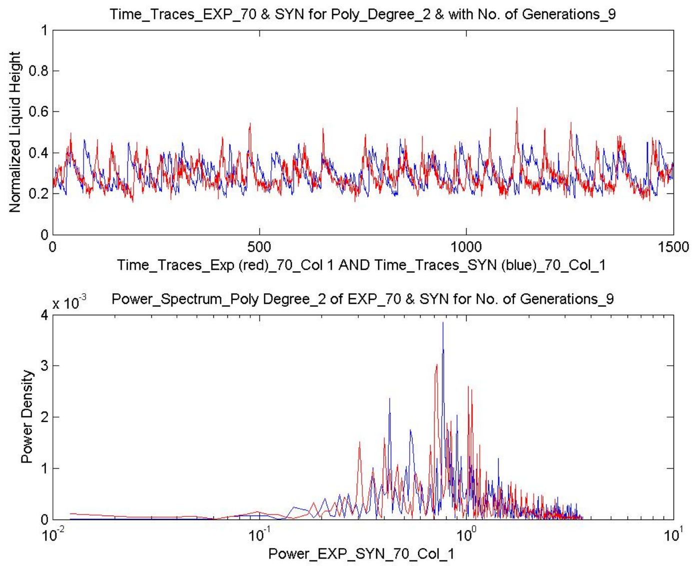


(f) ***Column Six***

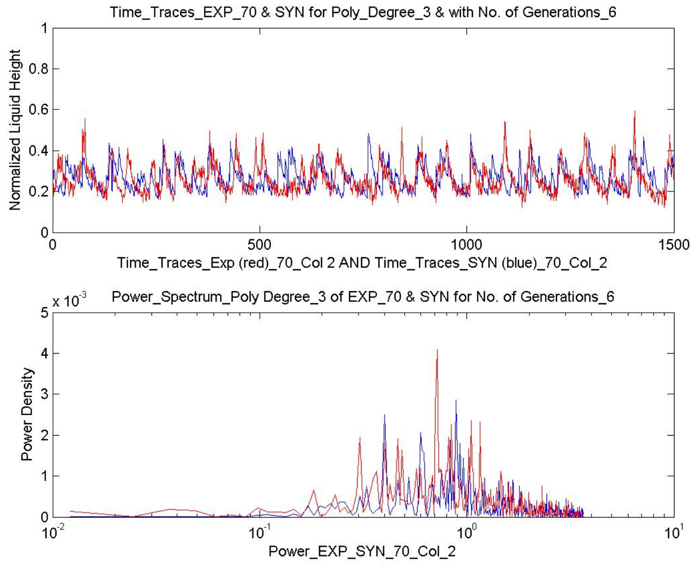


9. **EXOct16_160117_rn_70_w_0.474_o_0.000_g_3.972_bi_0.00_RW_**

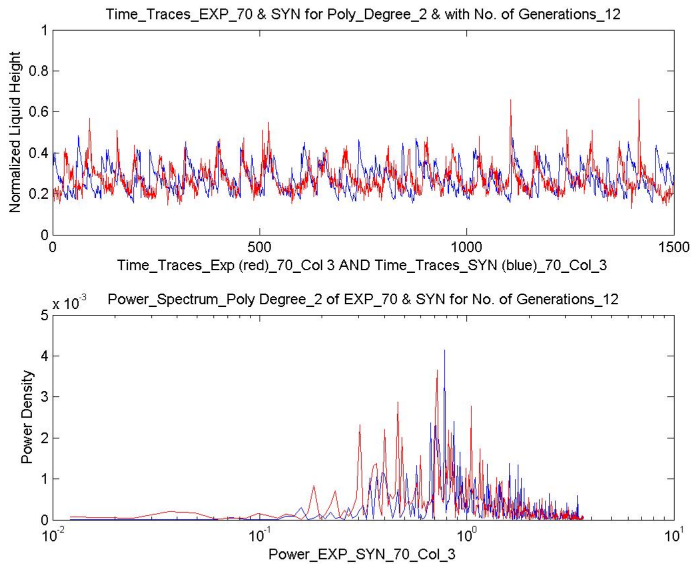
   (a) ***Column One***



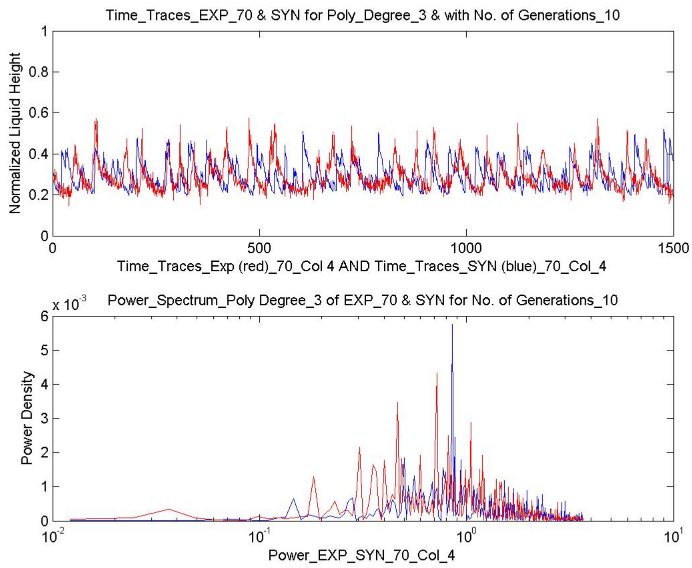
   (b) ***Column Two***



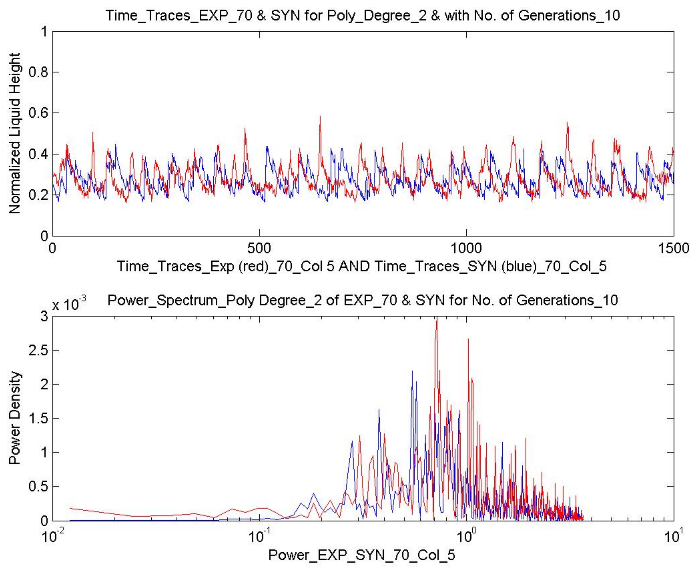
(c) ***Column Three***



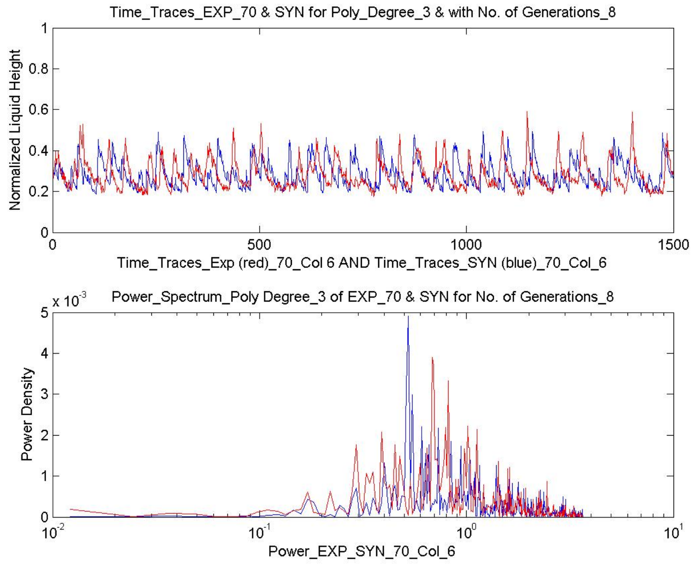
(d) ***Column Four***



(e) ***Column Five***

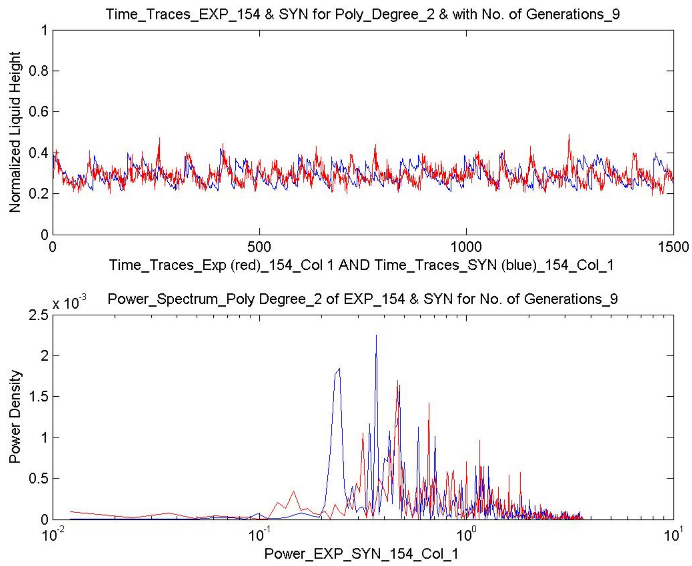


(f) ***Column Six***

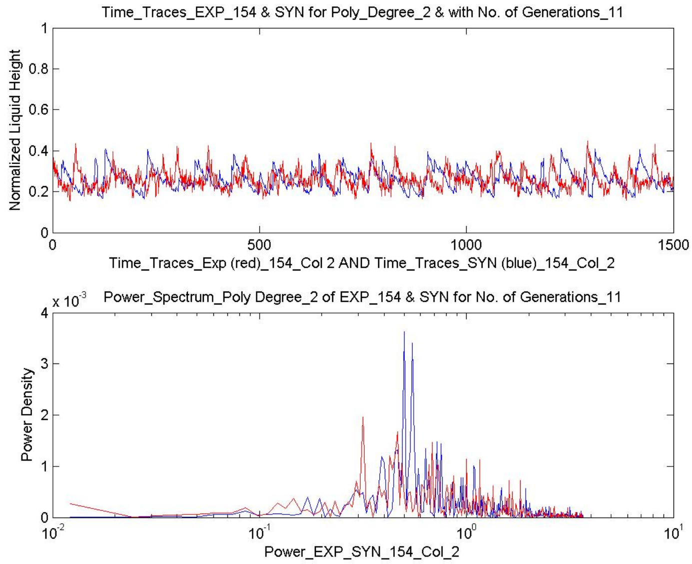


10. **EXOct22_130914_rn_154_w_0.321_o_0.000_g_3.399_bi_0.10_RW_**

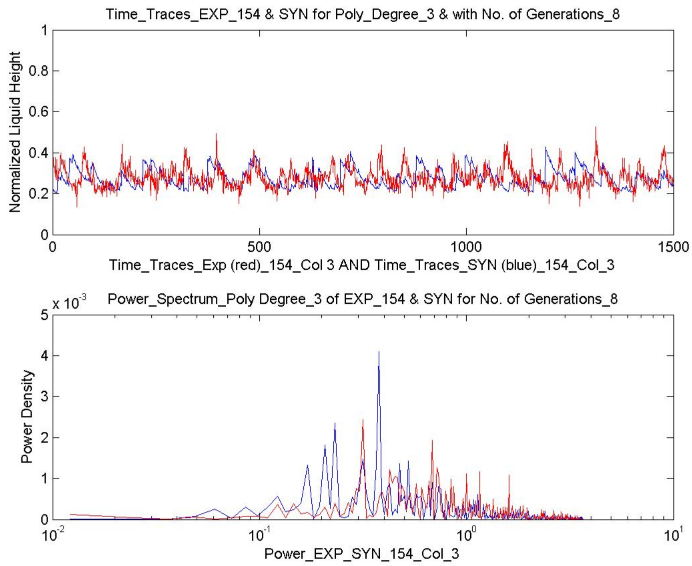
   (a) ***Column One***



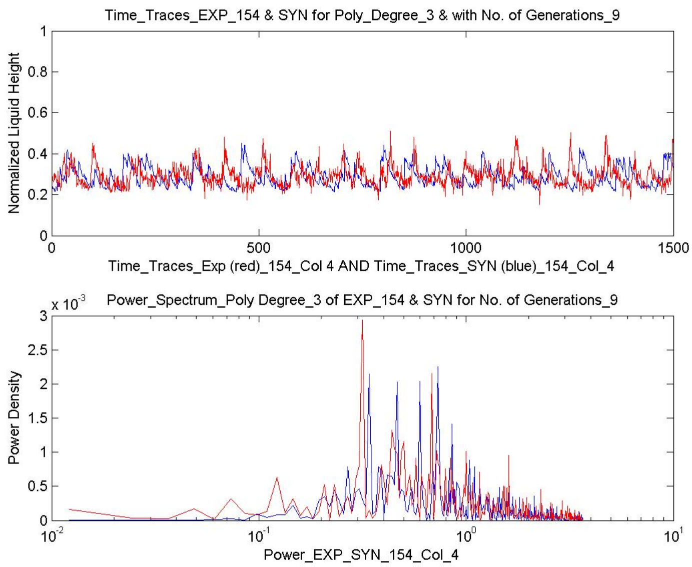
   (b) ***Column Two***



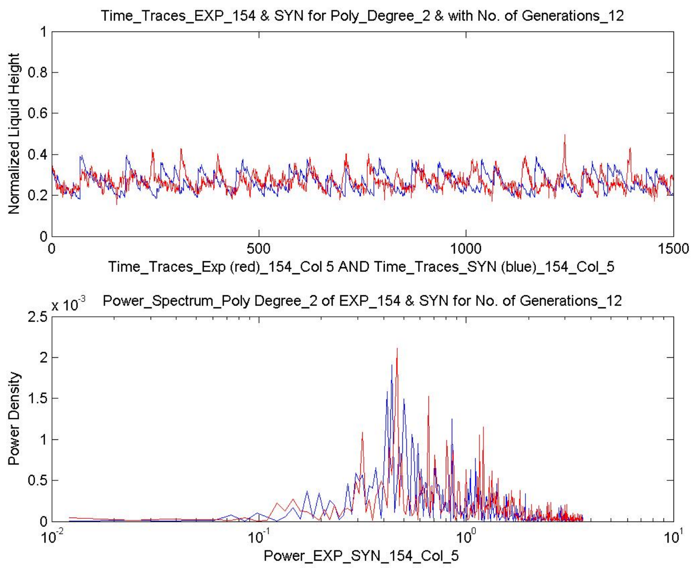
(c) ***Column Three***



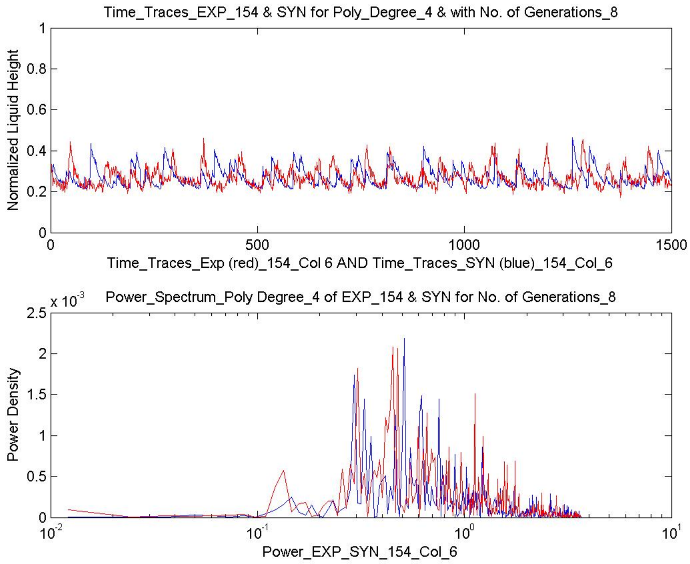
(d) ***Column Four***



(e) ***Column Five***

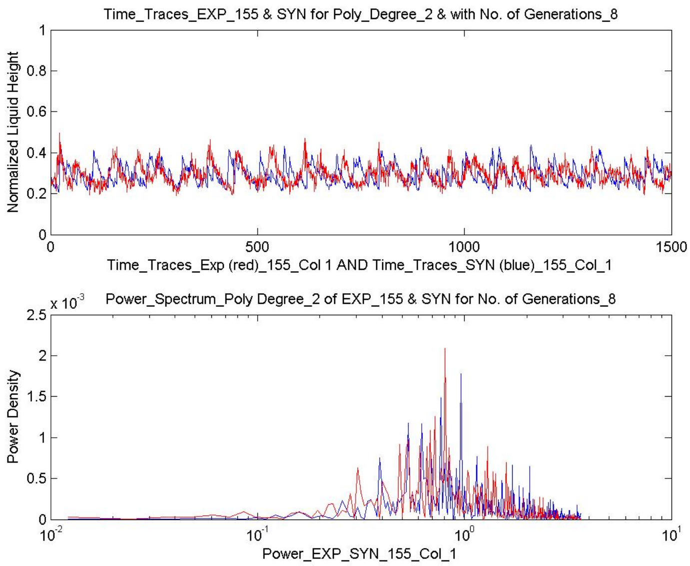


(f) ***Column Six***

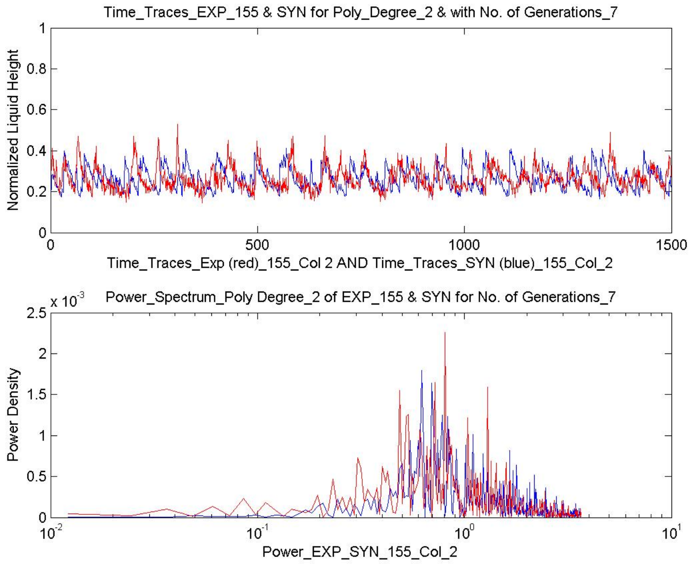


11. **EXOct22_130914_rn_155_w_0.342_o_0.000_g_3.398_bi_0.10_RW_**

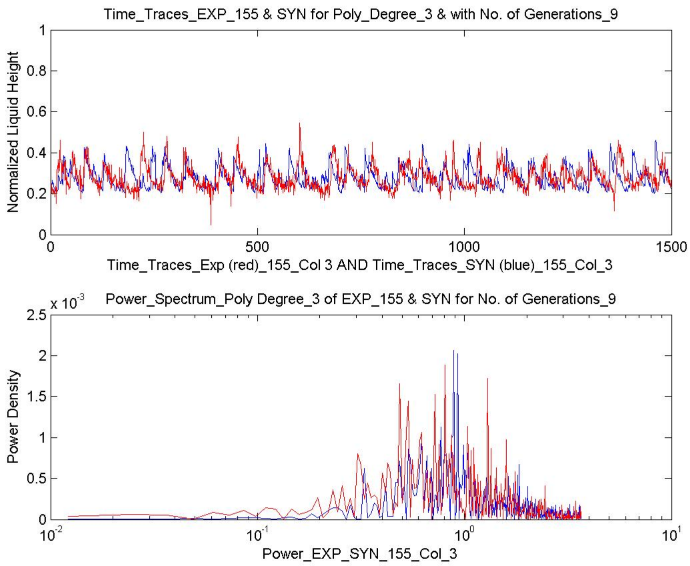
    (a) ***Column One***



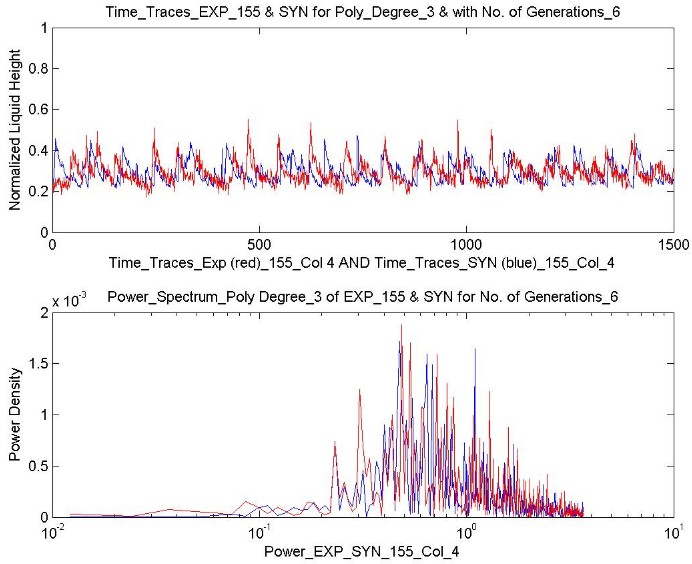
    (b) ***Column Two***



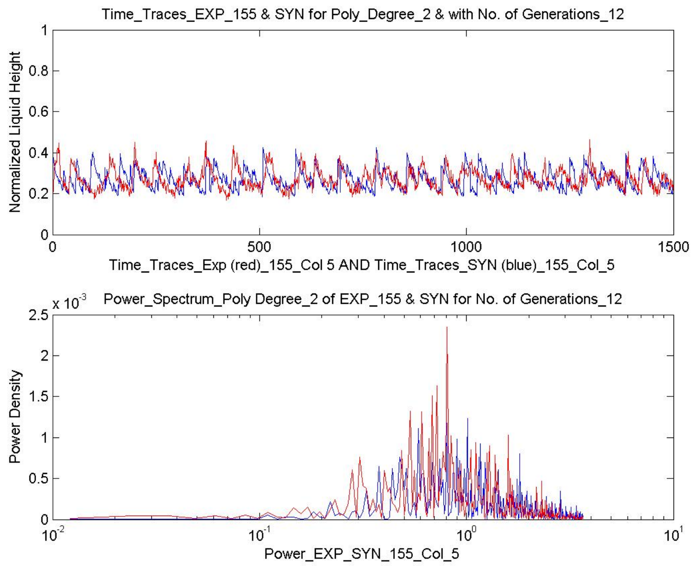
(c) ***Column Three***



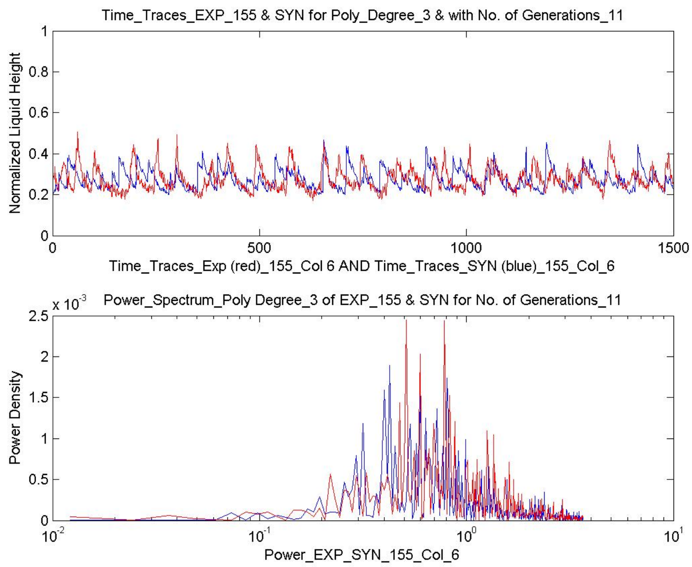
(d) ***Column Four***



(e) ***Column Five***

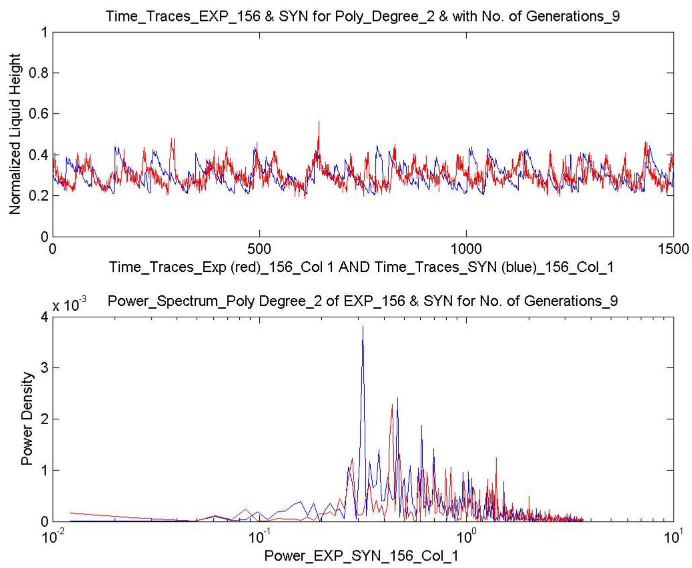


(f) ***Column Six***

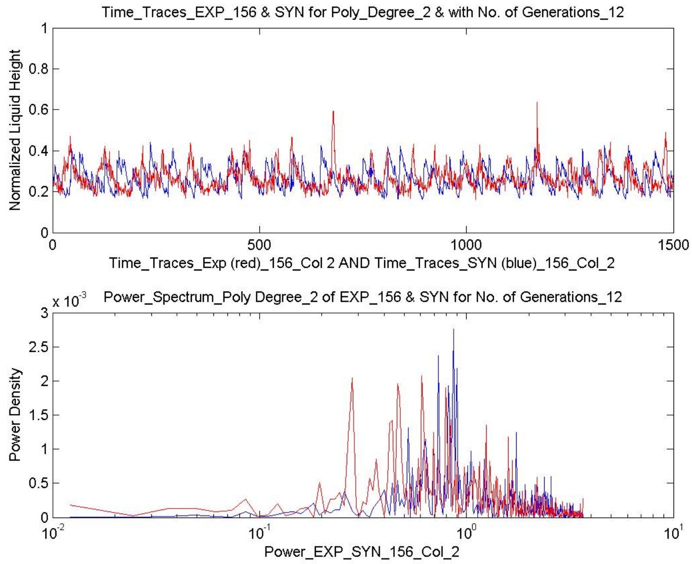


12. **EXOct22_130914_rn_156_w_0.362_o_0.000_g_3.393_bi_0.10_RW_**

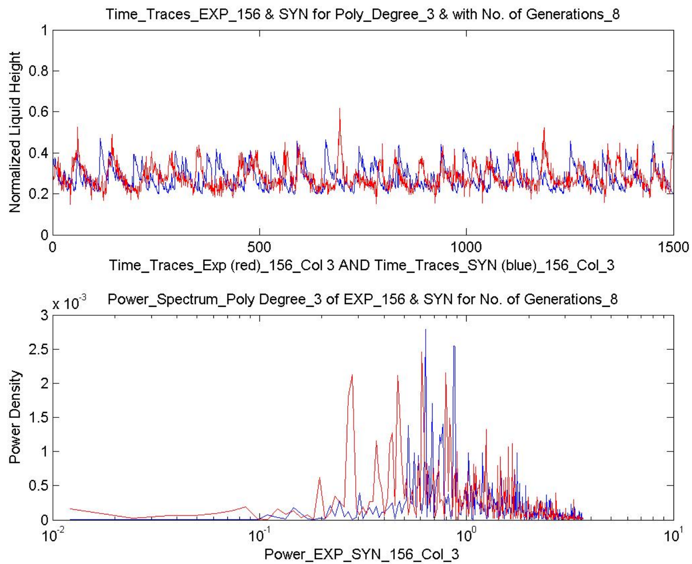
(a) ***Column One***



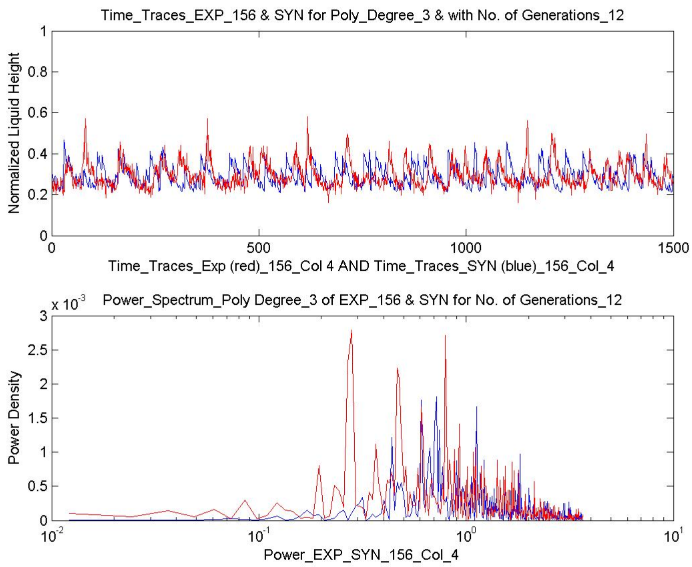
(b) ***Column Two***



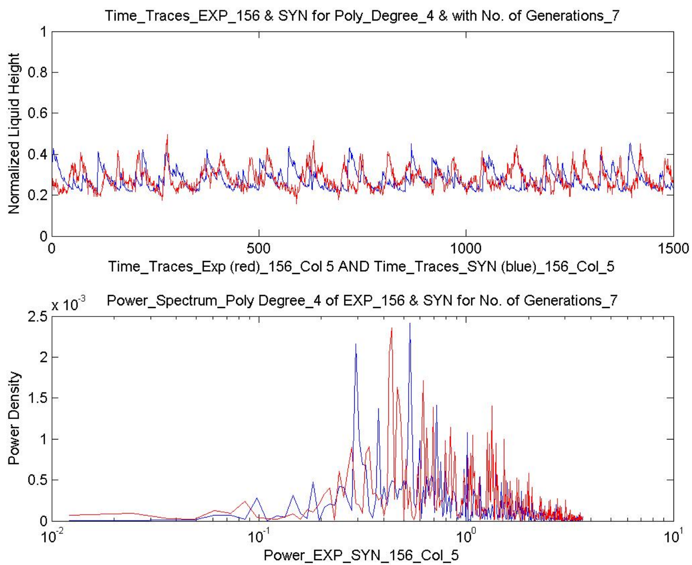
(c) ***Column Three***



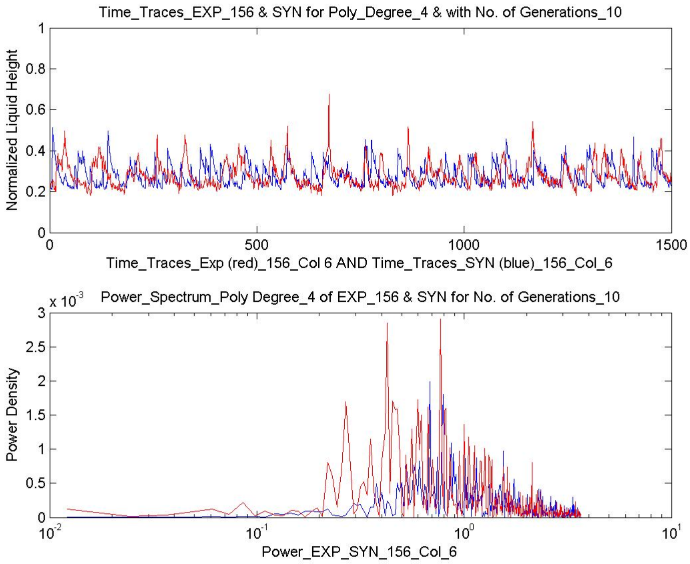
(d) ***Column Four***



(e) ***Column Five***

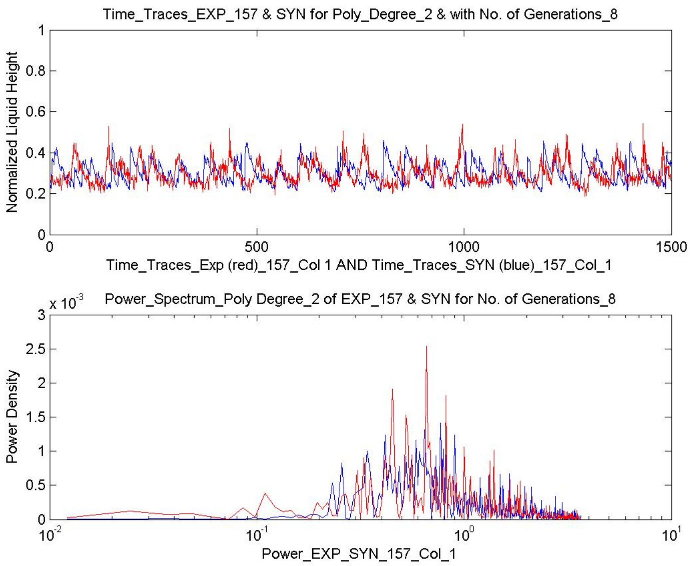


(f) ***Column Six***

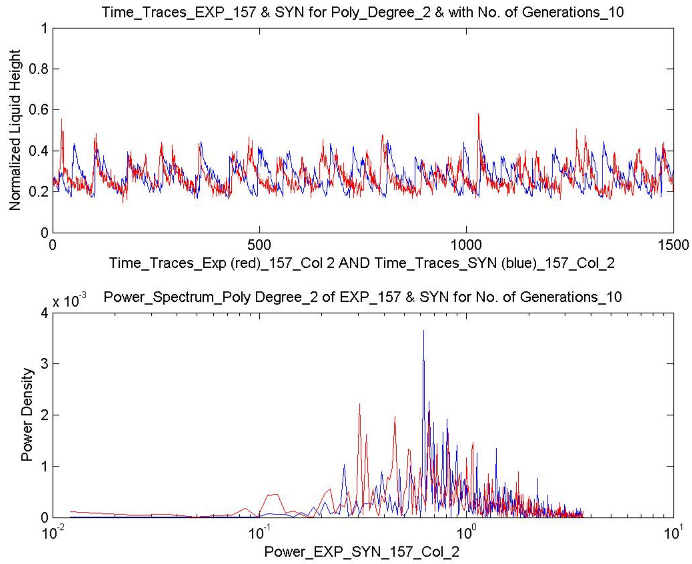


13. **EXOct22_130914_rn_157_w_0.383_o_0.000_g_3.389_bi_0.10_RW_**

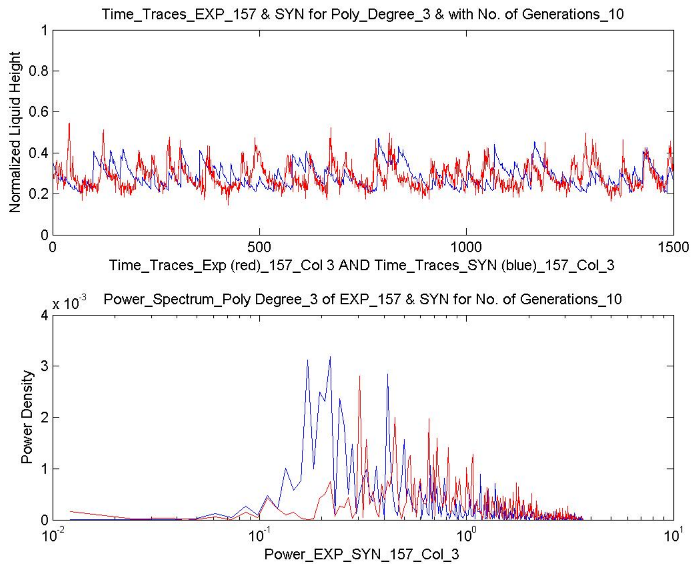
   (a) ***Column One***



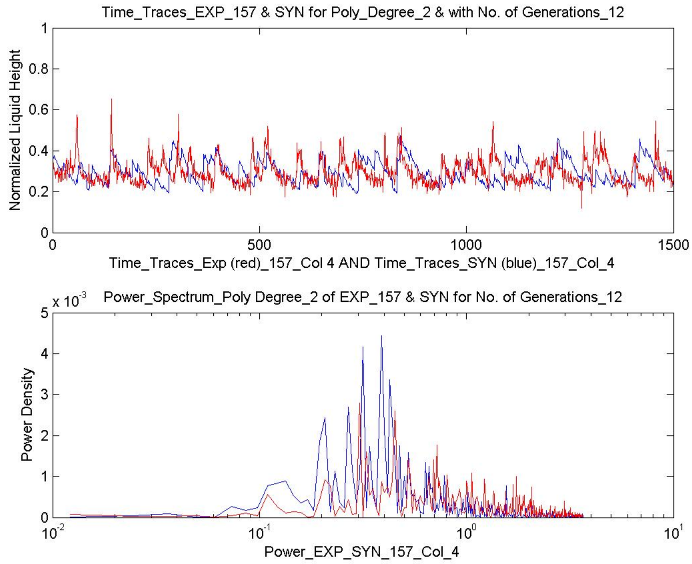
   (b) ***Column Two***



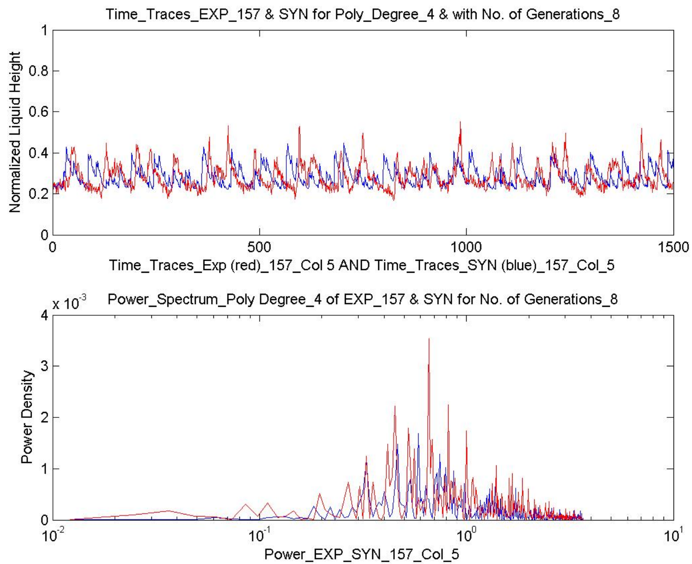
(c) ***Column Three***



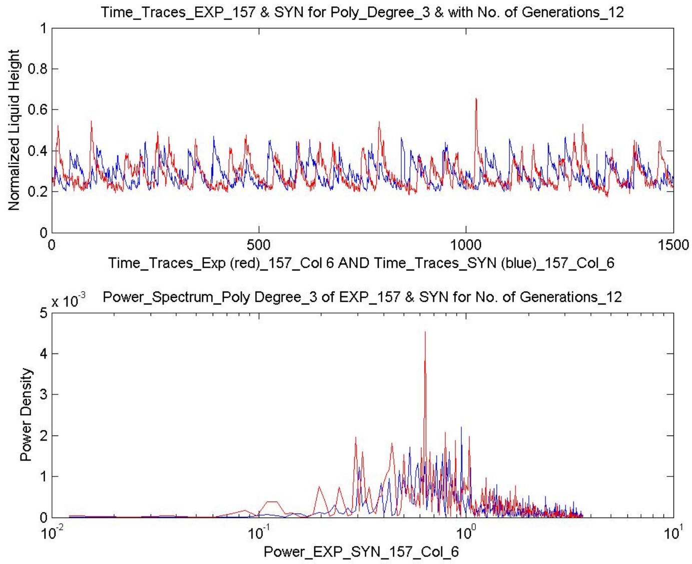
(d) ***Column Four***



(e) ***Column Five***

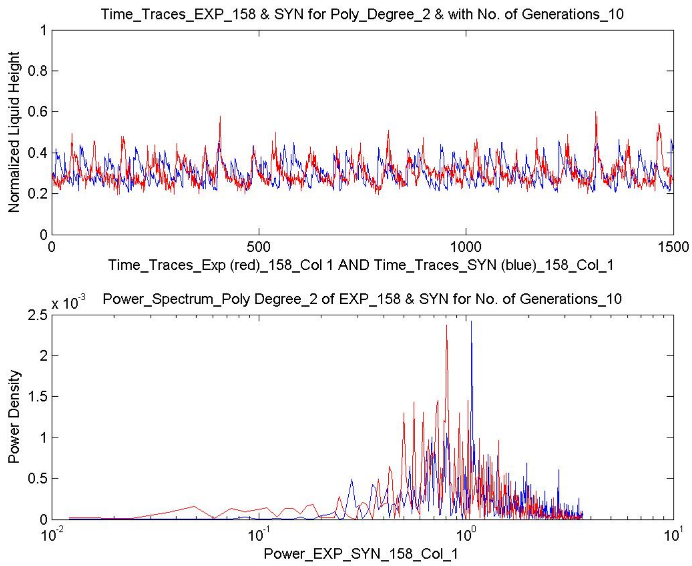


(f) ***Column Six***

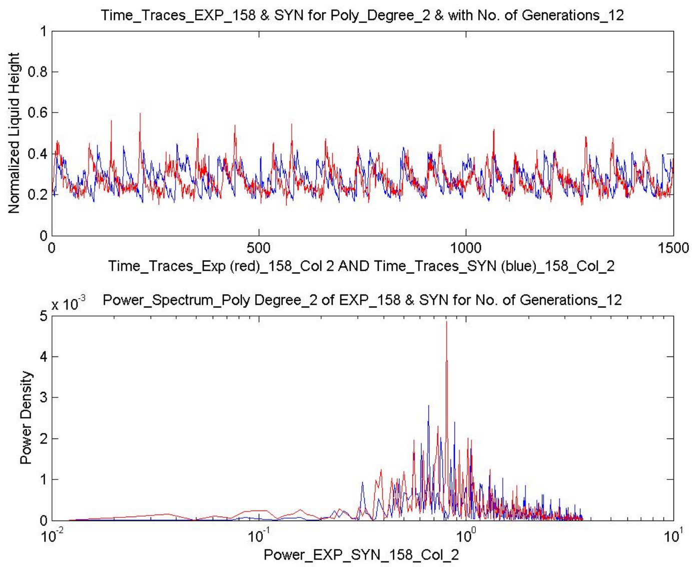


14. **EXOct22_130914_rn_158_w_0.402_o_0.000_g_3.397_bi_0.10_RW_**

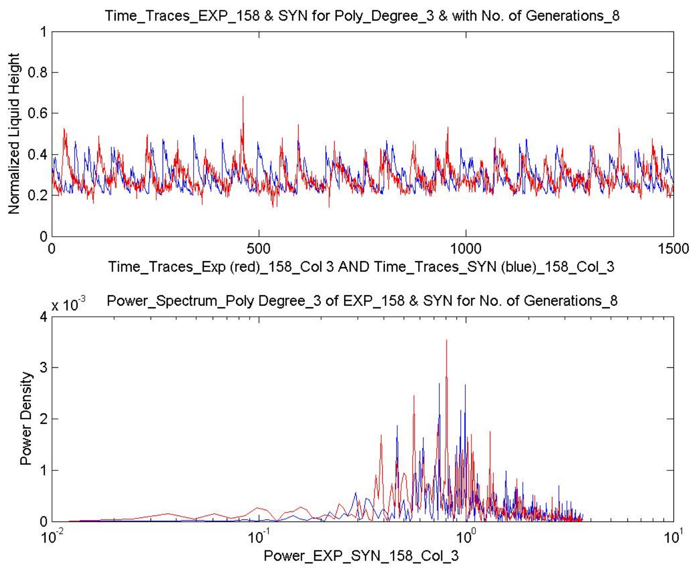
    (a) ***Column One***



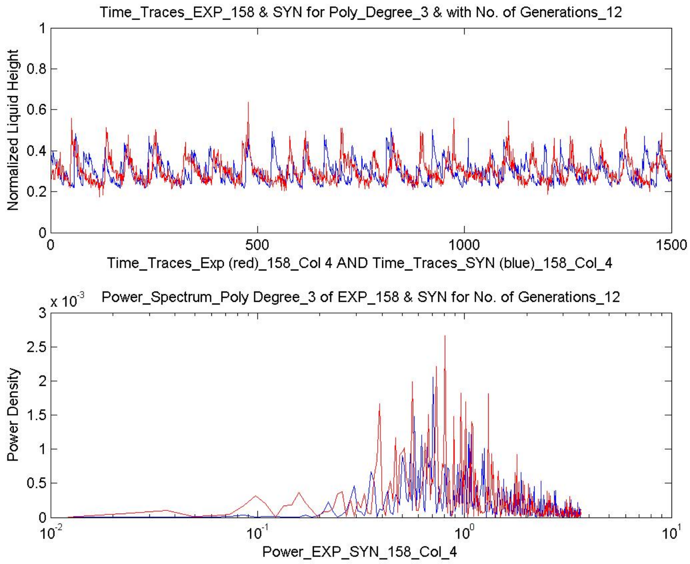
    (b) ***Column Two***



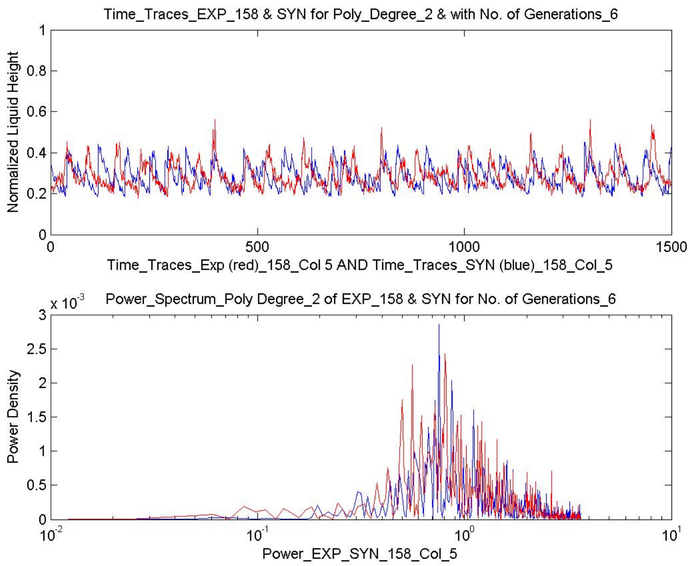
(c) ***Column Three***



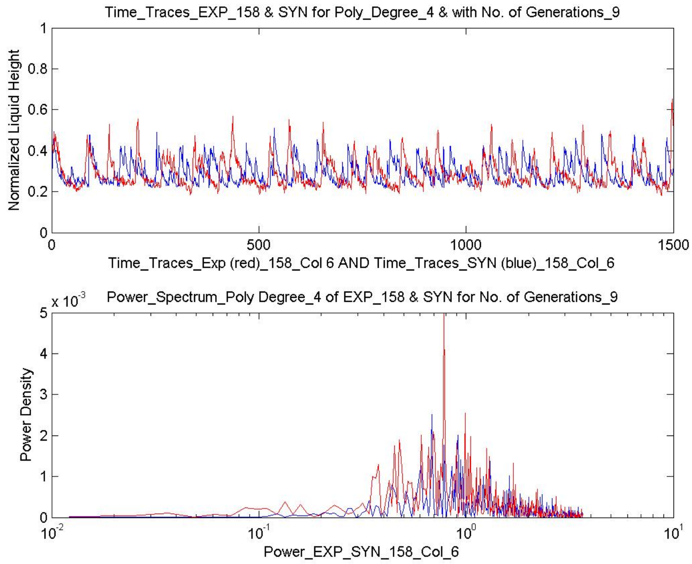
(d) ***Column Four***



(e) ***Column Five***

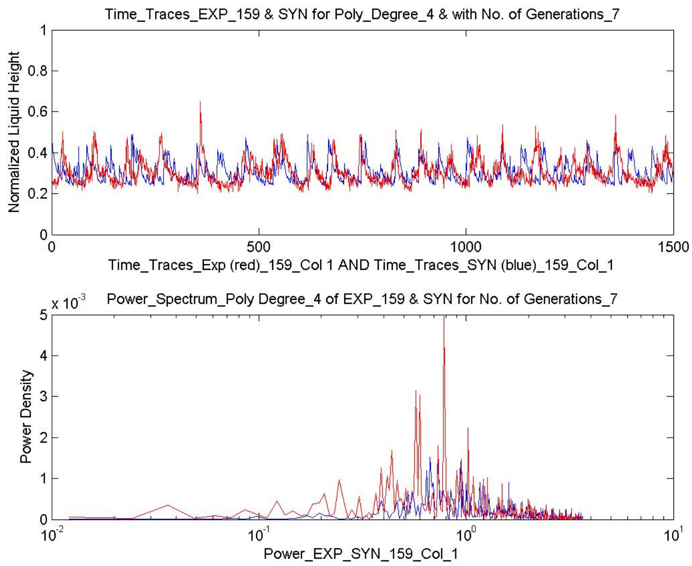


(f) ***Column Six***

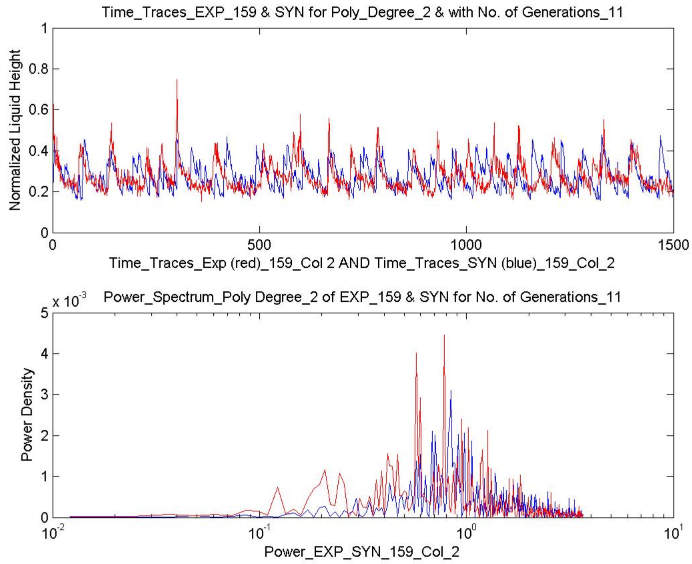


15. **EXOct22_130914_rn_159_w_0.421_o_0.000_g_3.394_bi_0.10_RW_**

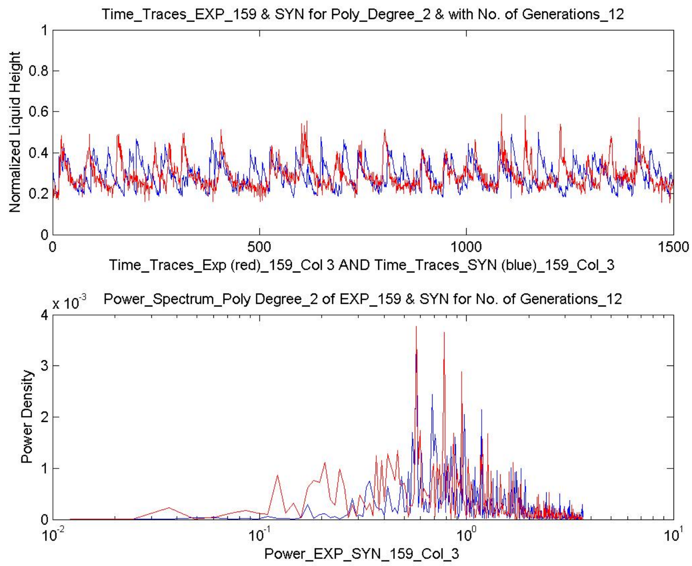
(a) ***Column One***



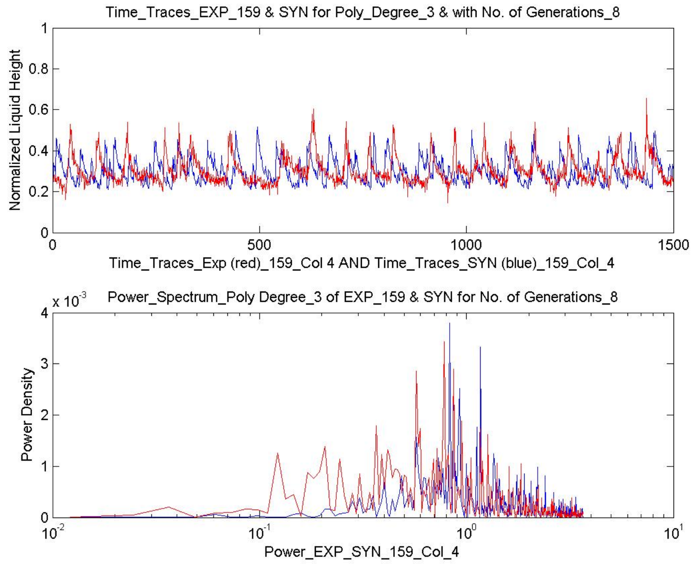
(b) ***Column Two***



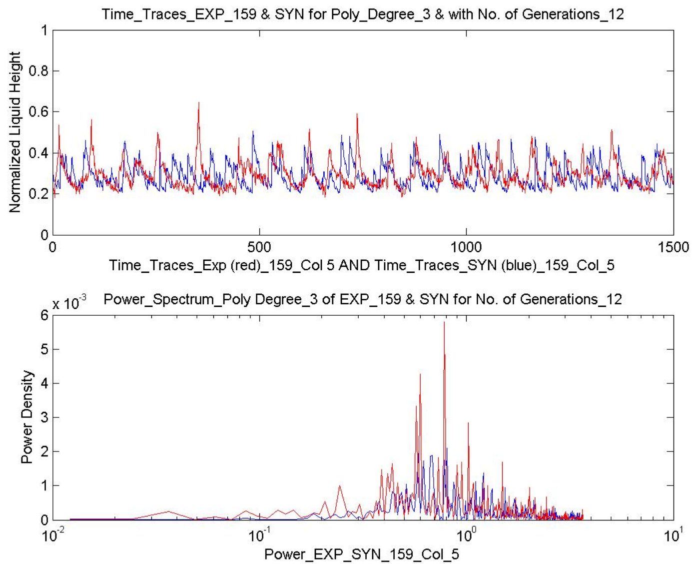
(c) ***Column Three***



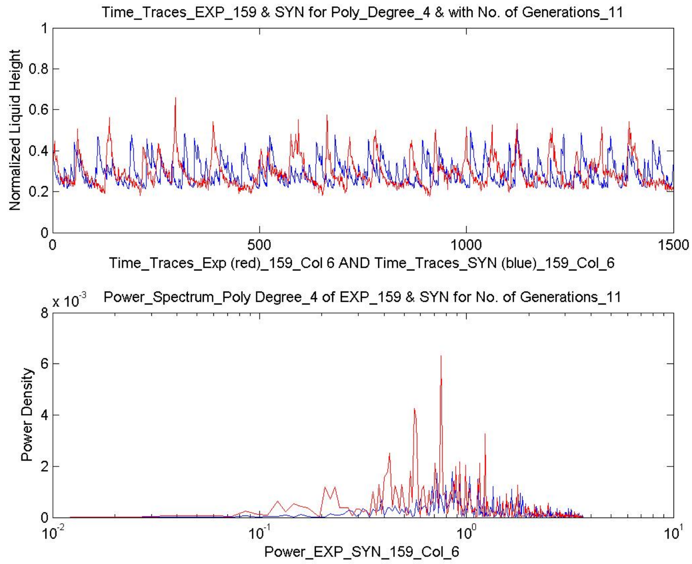
(d) ***Column Four***



(e) ***Column Five***

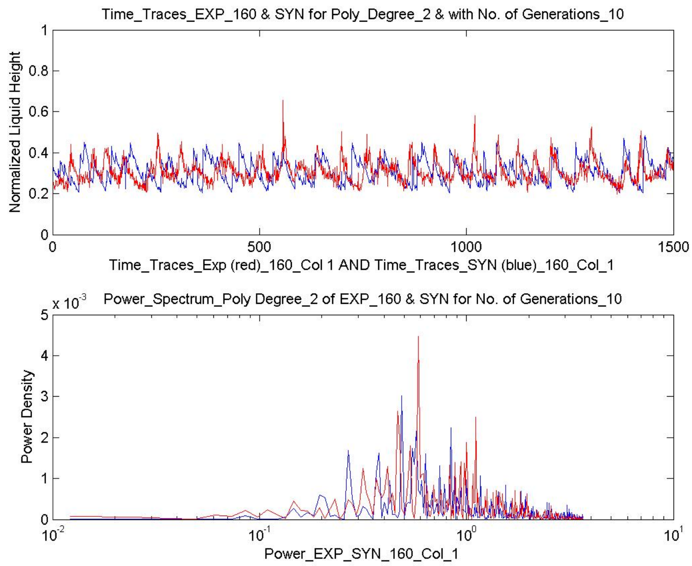


(f) ***Column Six***

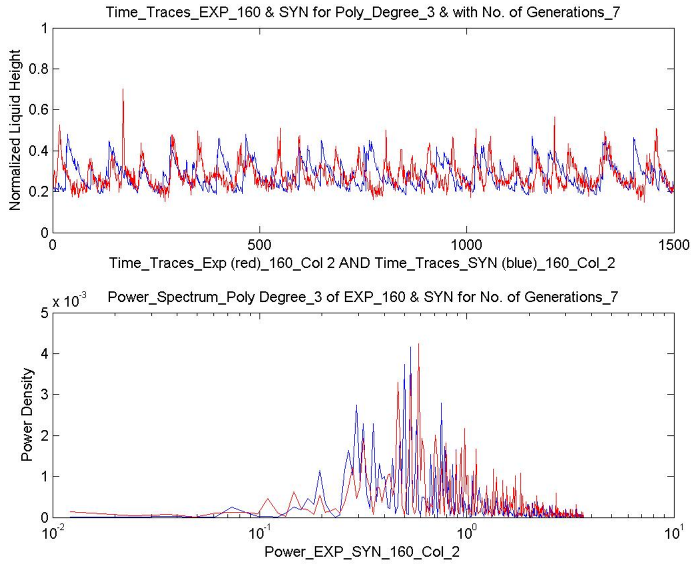


16. **EXOct22_130914_rn_160_w_0.440_o_0.000_g_3.389_bi_0.10_RW_**

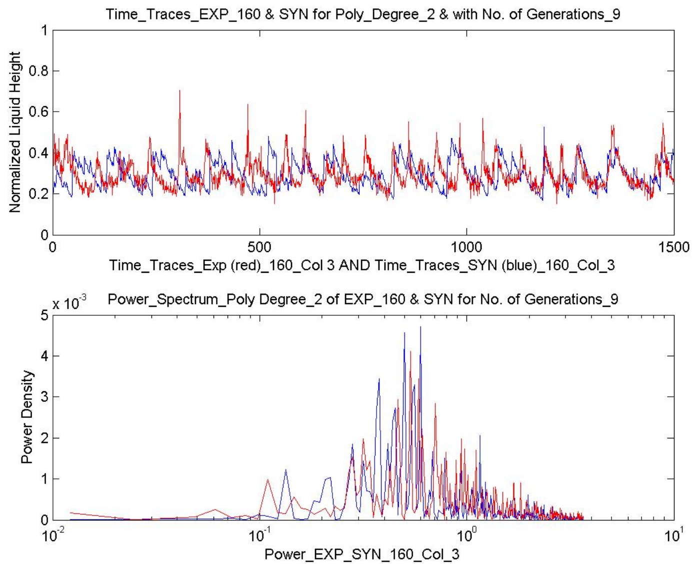
(a) ***Column One***



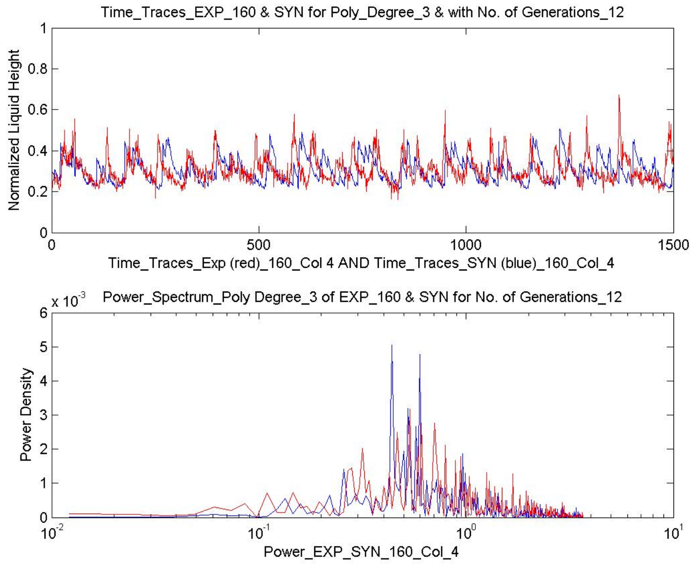
(b) ***Column Two***



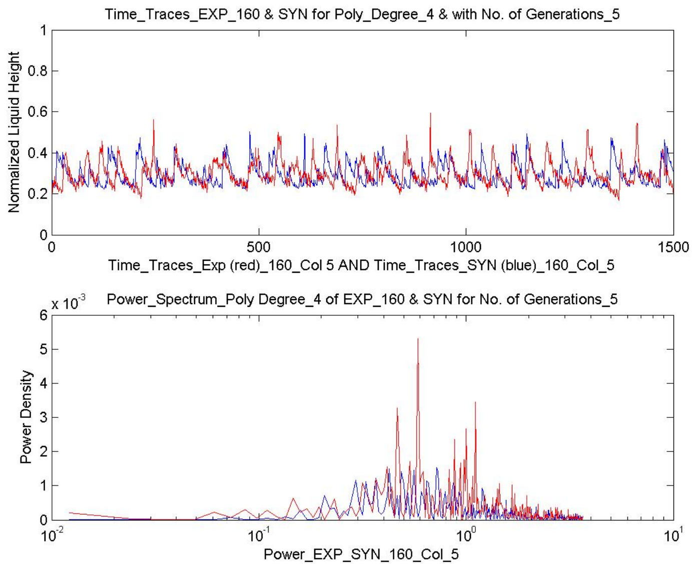
(c) ***Column Three***



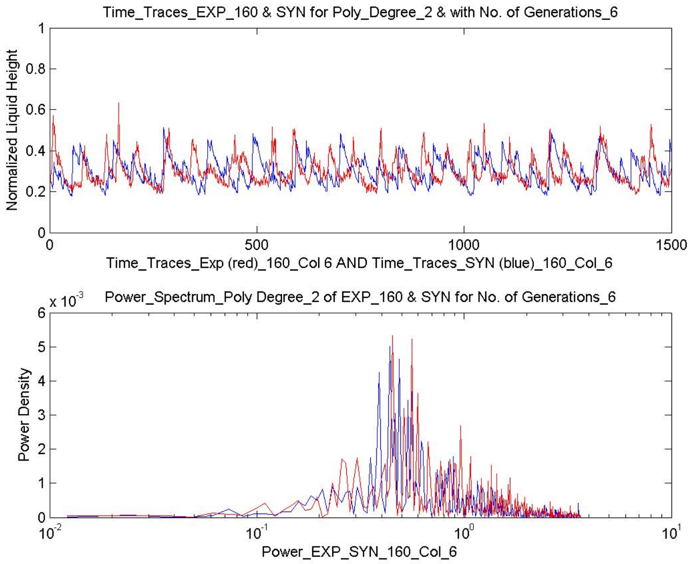
(d) ***Column Four***



(e) ***Column Five***

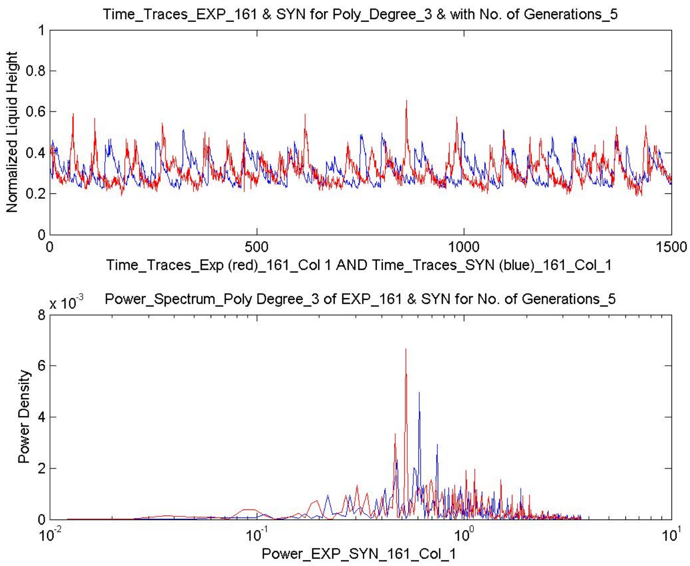


(f) ***Column Six***

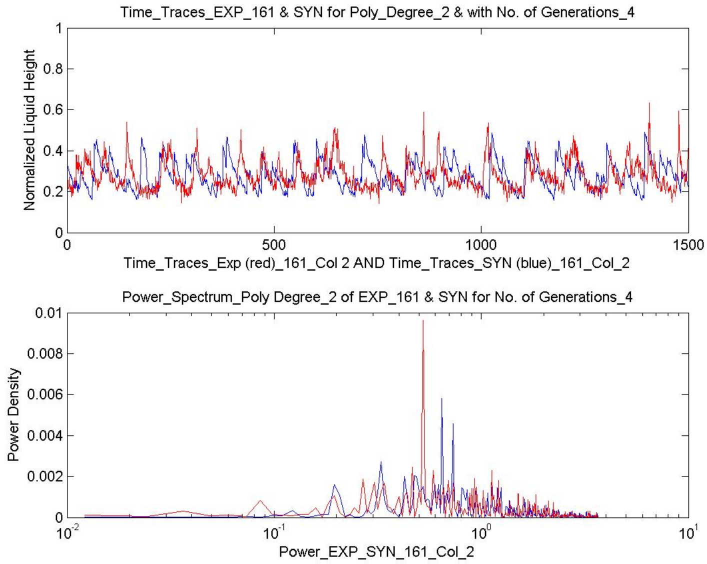


17. **EXOct22_130914_rn_161_w_0.462_o_0.000_g_3.393_bi_0.10_RW_**

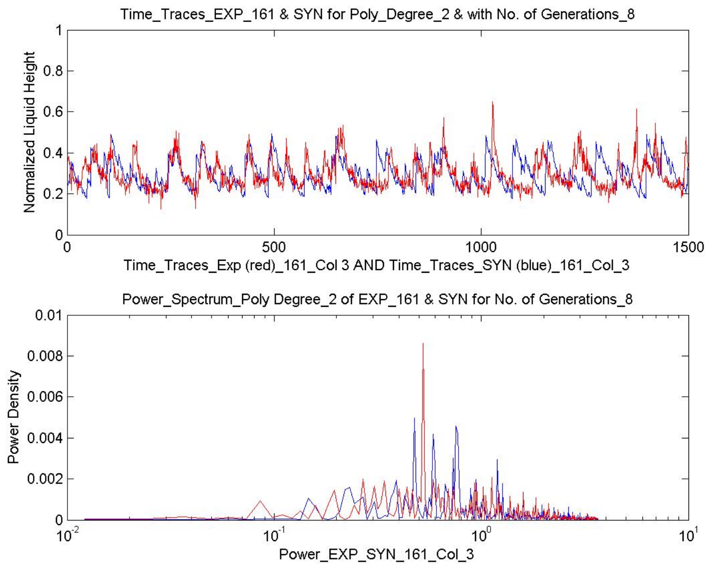
(a) ***Column One***



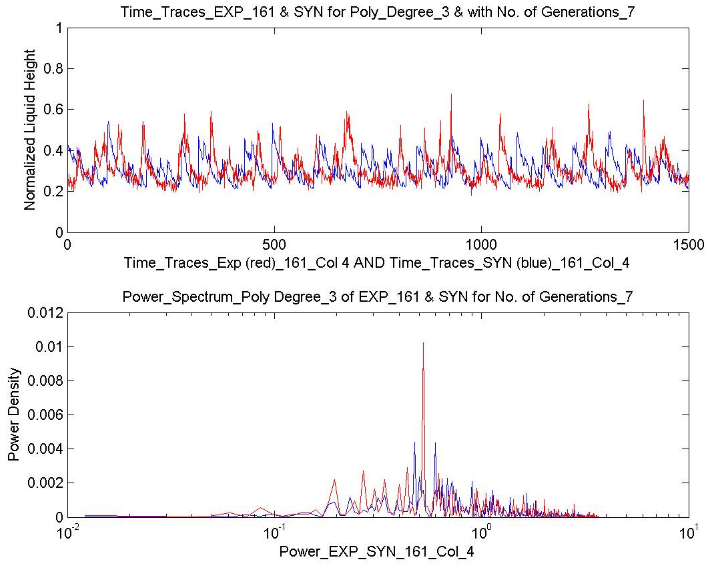
(b) ***Column Two***



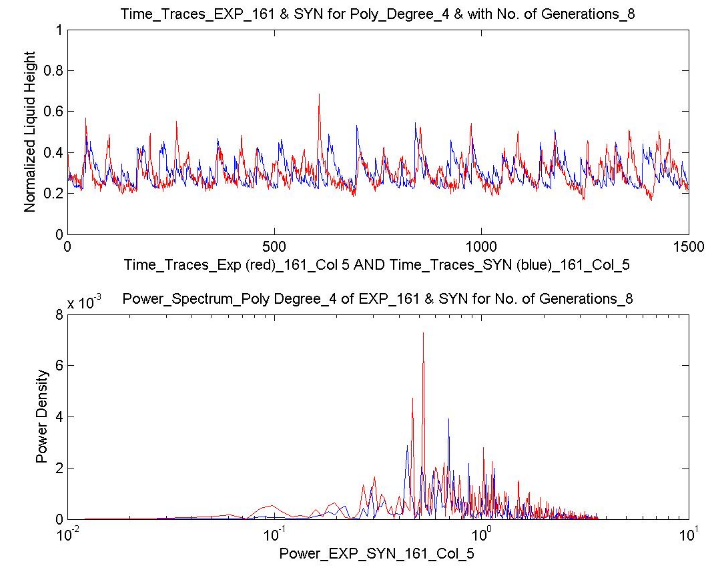
(c) ***Column Three***



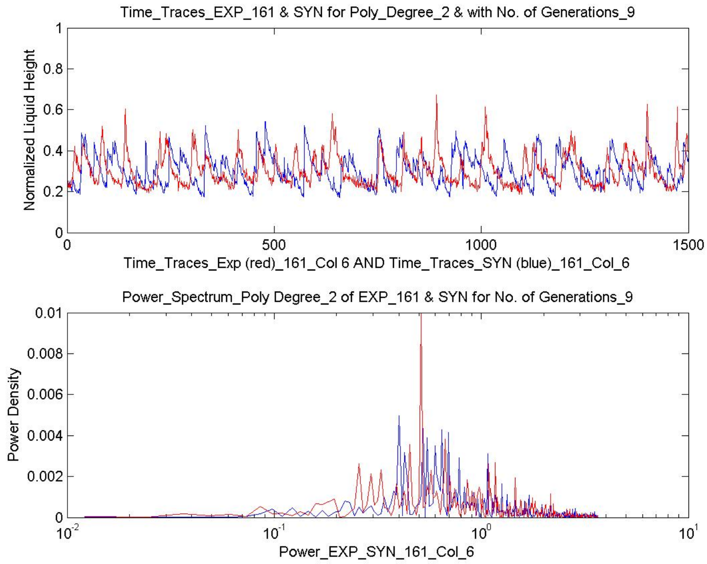
(d) ***Column Four***



(e) ***Column Five***



(f) ***Column Six***


Time_Traces_EXP_161 & SYN for Poly_Degree_2 & with No. of Generations_9
Normalized Liquid Height
Time_Traces_Exp (red)_161_Col 6 AND Time_Traces_SYN (blue)_161_Col_6
Power_Spectrum_Poly Degree_2 of EXP_161 & SYN for No. of Generations_9
Power Density
Power_EXP_SYN_161_Col_6

Appendix G

The Unsuccessful Sets

If the major frequency is not available, the algorithm will not function as there will not be enough parameters. Two of the data sets[1] [2] given are unable to be analyzed using this algorithm in the work as shown in the tables below.

Mean	STD	H_omega	Max	Min	MP*	Freq*	Nbox*
0.3542	0.0066	0.0187	0.4011	0.3095	0.0002	0.0122	1.0000
0.3468	0.0092	0.0260	0.3956	0.3468	0.0002	0.0122	1.0000
0.3398	0.0111	0.0314	0.4176	0.2623	0.0002	0.0122	1.0000
0.3312	0.0103	0.0291	0.4049	0.2562	0.0002	0.0122	1.0000
0.3228	0.0127	0.0359	0.3680	0.2667	0.0002	0.0122	1.0000
0.3242	0.0094	0.0266	0.3560	0.2906	0.0002	0.0122	1.0000

Figure G.1: *The EXOct15_93 measurement does not provide valid major power and its frequency*

Mean	STD	H_omega	Max	Min	MP*	Freq*	Nbox*
0.3305	0.0172	0.0486	0.4016	0.2700	0.0037	0.0122	1.0000
0.3491	0.0211	0.0597	0.4228	0.2923	0.0037	0.0122	1.0000
0.3533	0.0277	0.0783	0.4372	0.2546	0.0037	0.0122	1.0000
0.3546	0.0223	0.0631	0.4479	0.2443	0.0037	0.0122	1.0000
0.3037	0.0203	0.0574	0.3794	0.2253	0.0037	0.0122	1.0000
0.3257	0.0208	0.0588	0.3936	0.2756	0.0037	0.0122	1.0000

Figure G.2: *The EXOct15_94 measurement does not provide valid major power and its frequency*

[1] EXOct15_173335_rn_93_w_0.030_o_0.000_g_0.408_bi_0.00
[2] EXOct15_173335_rn_94_w_0.036_o_0.000_g_0.866_bi_0.00

Bibliography

[1] John A Rice, Mathematical Statistics and Data Analysis, 2nd edition, Wadsworth Inc. 1995

[2] Jens Feder, Fractals, Fourth Printing, Plenum Press, New York, 1989

[3] , McMellan, Yoder, Schafer, Signal Processing First, First edition,Prentice Hall, 2003

[4] MATLAB Online Resources, Mathworlds, MATLAB Manual

[5] Clifford Brown, Larry Liebovitch, Fractal Analysis, SAGE, 2010